logisland Documentation

Release 1.1.2

bailet.thomas

Jan 24, 2023

Contents

Chat with us on Gitter

Download the latest release build and unzip on an edge node.

CHAPTER 1

Contents:

1.1 User Documentation

Contents:

1.1.1 Components

Contents:

Engines Documentation

Contents:

Engine-spark

Find below the list.

ConsoleStructuredStreamProviderService

Provide a ways to print output in console in a StructuredStream streams

Class

com.hurence.logisland.stream.spark.structured.provider.ConsoleStructuredStreamProviderService

Tags

None.

Properties

This component has no required or optional properties.

Extra informations

No additional information is provided

DummyRecordStream

No description provided.

Class

com.hurence.logisland.stream.spark.DummyRecordStream

Tags

None.

Properties

This component has no required or optional properties.

Extra informations

No additional information is provided

KafkaConnectBaseProviderService

No description provided.

Class

com.hurence.logisland.stream.spark.provider.KafkaConnectBaseProviderService

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kc.connector.class	The class canonical name of the kafka connector to use.		null	false	false
kc.connector.properties	The properties (key=value) for the connector.			false	false
kc.data.key.converter	Key converter class		null	false	false
kc.data.key.converter.properties	Key converter properties			false	false
kc.data.value.converter	Value converter class		null	false	false
kc.data.value.converter.properties	Value converter properties			false	false
kc.worker.tasks.max	Max number of threads for this connector		1	false	false
kc.partitions.max	Max number of partitions for this connector.		null	false	false
kc.connector.offset.backing.store	The backing store to be used.	memory (Standalone in memory offset backing store. Not suitable for clustered deployments unless source is unique or stateless), file (Standalone filesystem based offset backing store. You have to specify the property offset.storage.file.filename for the file path.Not suitable for clustered deployments unless source is unique or standalone), kafka (Distributed kafka topic based offset backing store. See the javadoc of class org.apache.kafka.connect.storage.KafkaOffsetBackingStore for the configuration options.This backing store is well suited for distributed deployments.)	memory	false	false
kc.connector.offset.backing.store.properties	Properties to configure the offset backing store			false	false

Extra informations

No additional information is provided

KafkaConnectStructuredSinkProviderService

No description provided.

Class

com.hurence.logisland.stream.spark.provider.KafkaConnectStructuredSinkProviderService

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kc.connector.class	The class canonical name of the kafka connector to use.		null	false	false
kc.connector.properties	The properties (key=value) for the connector.			false	false
kc.data.key.converter	Key converter class		null	false	false
kc.data.key.converter.properties	Key converter properties			false	false
kc.data.value.converter	Value converter class		null	false	false
kc.data.value.converter.properties	Value converter properties			false	false
kc.worker.tasks.max	Max number of threads for this connector		1	false	false
kc.partitions.max	Max number of partitions for this connector.		null	false	false
kc.connector.offset.backing.store	The backing store backing store to be used.	memory (Standalone in memory offset backing store. Not suitable for clustered deployments unless source is unique or stateless), file (Standalone filesystem based offset backing store. You have to specify the property offset.storage.file.filename for the file path.Not suitable for clustered deployments unless source is unique or standalone), kafka (Distributed kafka topic based offset backing store. See the javadoc of class org.apache.kafka.connect.storage.KafkaOffsetBackingStore for the configuration options.This backing store is well suited for distributed deployments.)	memory	false	false
kc.connector.offset.backing.store.properties	Properties to configure the offset backing store			false	false

Extra informations

No additional information is provided

KafkaConnectStructuredSourceProviderService

No description provided.

Class

com.hurence.logisland.stream.spark.provider.KafkaConnectStructuredSourceProviderService

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 3: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kc.connector.class	The class canonical name of the kafka connector to use.		null	false	false
kc.connector.properties	The properties (key=value) for the connector.			false	false
kc.data.key.converter	Key converter class		null	false	false
kc.data.key.converter.properties	Key converter properties			false	false
kc.data.value.converter	Value converter class		null	false	false
kc.data.value.converter.properties	Value converter properties			false	false
kc.worker.tasks.max	Max number of threads for this connector		1	false	false
kc.partitions.max	Max number of partitions for this connector.		null	false	false
kc.connector.offset.backing.store	The backing store backing store to be used.	memory (Standalone in memory offset backing store. Not suitable for clustered deployments unless source is unique or stateless), file (Standalone filesystem based offset backing store. You have to specify the property offset.storage.file.filename for the file path.Not suitable for clustered deployments unless source is unique or standalone), kafka (Distributed kafka topic based offset backing store. See the javadoc of class org.apache.kafka.connect.storage.KafkaOffsetBackingStore for the configuration options.This backing store is well suited for distributed deployments.)	memory	false	false
kc.connector.offset.backing.store.properties	Properties to configure the offset backing store			false	false

Extra informations

No additional information is provided

KafkaRecordStreamDebugger

No description provided.

Class

com.hurence.logisland.stream.spark.KafkaRecordStreamDebugger

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 4: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kafka.error.topics	Sets the error topics Kafka topic name		_errors	false	false
kafka.input.topics	Sets the input Kafka topic name		_raw	false	false
kafka.output.topics	Sets the output Kafka topic name		_records	false	false
avro.input.schema	the avro schema definition		null	false	false
avro.output.schema	the avro schema definition for the output serialization		null	false	false
kafka.input.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.output.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.error.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as	com.hurence.logisland.serializer.JsonSerializer	false	false

Extra informations

No additional information is provided

KafkaRecordStreamHDFSBurner

No description provided.

Class

com.hurence.logisland.stream.spark.KafkaRecordStreamHDFSBurner

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 5: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kafka.error.topics	Sets the error topics Kafka topic name		_errors	false	false
kafka.input.topics	Sets the input Kafka topic name		_raw	false	false
kafka.output.topics	Sets the output Kafka topic name		_records	false	false
avro.input.schema	the avro schema definition		null	false	false
avro.output.schema	the avro schema definition for the output serialization		null	false	false
kafka.input.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.output.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.error.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as	com.hurence.logisland.serializer.JsonSerializer	false	false

Extra informations

No additional information is provided

KafkaRecordStreamParallelProcessing

No description provided.

Class

com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 6: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kafka.error.topics	Sets the error topics Kafka topic name		_errors	false	false
kafka.input.topics	Sets the input Kafka topic name		_raw	false	false
kafka.output.topics	Sets the output Kafka topic name		_records	false	false
avro.input.schema	the avro schema definition		null	false	false
avro.output.schema	the avro schema definition for the output serialization		null	false	false
kafka.input.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.output.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.error.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as	com.hurence.logisland.serializer.JsonSerializer	false	false

Extra informations

No additional information is provided

KafkaRecordStreamSQLAggregator

This is a stream capable of SQL query interpretations.

Class

com.hurence.logisland.stream.spark.KafkaRecordStreamSQLAggregator

Tags

stream, SQL, query, record

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 7: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kafka.error.topics	Sets the error topics Kafka topic name		_errors	false	false
kafka.input.topics	Sets the input Kafka topic name		_raw	false	false
kafka.output.topics	Sets the output Kafka topic name		_records	false	false
avro.input.schema	the avro schema definition		null	false	false
avro.output.schema	the avro schema definition for the output serialization		null	false	false
kafka.input.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.output.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.error.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as	com.hurence.logisland.serializer.JsonSerializer	false	false

Extra informations

No additional information is provided

KafkaStreamProcessingEngine

No description provided.

Class

com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 8: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
spark.app.name	The application name		logisland	false	false
spark.master	The url to Spark Master		local[2]	false	false
spark.monitoring.driver.port	The port for exposing monitoring metrics		null	false	false
spark.yarn.deploy-mode	The yarn deploy mode		null	false	false
spark.yarn.queue	The name of the YARN queue		default	false	false
spark.driver.memory	The memory size for Spark driver		512m	false	false
spark.executor.memory	The memory size for Spark executors		1g	false	false
spark.driver.cores	The number of cores for Spark driver		4	false	false
spark.executor.cores	The number of cores for Spark driver		1	false	false
spark.executor.instances	The number of instances for Spark app		null	false	false
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form		org.apache.spark.serializer.KryoSerializer	false	false
spark.streaming.blockInterval	Interval at which data received by Spark Streaming receivers is chunked into blocks of data before storing them in Spark. Minimum recommended - 50 ms		350	false	false
spark.streaming.kafka.maxRatePerPartition	Maximum rate (number of records per second) at which data will be read from each Kafka partition		5000	false	false
spark.streaming.batchDuration	Number of seconds per batch		2000	false	false

Continued on next page

Table 8 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
spark.streaming.backpressure.enabled	This enables the Spark Streaming to control the receiving rate based on the current batch scheduling delays and processing times so that the system receives only as fast as the system can process.		false	false	false
spark.streaming.unpersist	Force RDDs generated and persisted by Spark Streaming to be automatically unpersisted from Spark's memory. The raw input data received by Spark Streaming is also automatically cleared. Setting this to false will allow the raw data and persisted RDDs to be accessible outside the streaming application as they will not be cleared automatically. But it comes at the cost of higher memory usage in Spark.		false	false	false
spark.ui.port	No Description Provided.		4050	false	false
spark.streaming.kafka.maxRetries	No Description Provided.		-1	false	false
spark.streaming.kafka.maxRatePerPartition	Maximum rate (number of records per second) at which data will be read from each Kafka partition		3	false	false
spark.streaming.ui.retainedBatches	How many batches the Spark Streaming UI and status APIs remember before garbage collecting.		200	false	false
spark.streaming.receiver.writeAheadLog.enable	Enable write ahead logs for receivers. All the input data received through receivers will be saved to write ahead logs that will allow it to be recovered after driver failures.		false	false	false
spark.yarn.maxAppAttempts	Because Spark driver and Application Master share a single JVM, any error in Spark driver stops our long-running job. Fortunately it is possible to configure maximum number of attempts that will be made to re-run the application. It is reasonable to set higher value than default 2 (derived from YARN cluster property yarn.resourcemanager.am.max-attempts). 4 works quite well, higher value may cause unnecessary restarts even if the reason of the failure is permanent.		4	false	false
spark.yarn.am.attemptFailuresValidityInterval	If the application runs for days or weeks without restart or redeployment on highly utilized cluster, 4 attempts could be exhausted in few hours. To avoid this situation, the attempt counter should be reset on every hour of so.		1h	false	false

Continued on next page

Table 8 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
spark.yarn.max.executor.failures	maximum number of executor failures before the application fails. By default it is max(2 * num executors, 3), well suited for batch jobs but not for long-running jobs. The property comes with corresponding validity interval which also should be set.8 * num_executors		20	false	false
spark.yarn.executor.failuresValidityInterval	If the application runs for days or weeks without restart or redeployment on highly utilized cluster, x attempts could be exhausted in few hours. To avoid this situation, the attempt counter should be reset on every hour of so.		1h	false	false
spark.task.maxFailures	For long-running jobs you could also consider to boost maximum number of task failures before giving up the job. By default tasks will be retried 4 times and then job fails.		8	false	false
spark.memory.fraction	expresses the size of M as a fraction of the (JVM heap space - 300MB) (default 0.75). The rest of the space (25%) is reserved for user data structures, internal metadata in Spark, and safeguarding against OOM errors in the case of sparse and unusually large records.		0.6	false	false
spark.memory.storageFraction	expresses the size of R as a fraction of M (default 0.5). R is the storage space within M where cached blocks immune to being evicted by execution.		0.5	false	false
spark.scheduler.mode	The scheduling mode between jobs submitted to the same SparkContext. Can be set to FAIR to use fair sharing instead of queueing jobs one after another. Useful for multi-user services.	FAIR (fair sharing), FIFO (queueing jobs one after another)	FAIR	false	false
spark.properties.file	Load using –properties-file option while submitting spark job		null	false	false
java.library.path	The java library path to use with mesos.		null	false	false
spark.cores.max	The maximum number of total executor core with mesos.		null	false	false

Extra informations

No additional information is provided

KafkaStructuredStreamProviderService

Provide a ways to use kafka as input or output in StructuredStream streams

Class

com.hurence.logisland.stream.spark.structured.provider.KafkaStructuredStreamProviderService

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 9: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
kafka.error.topics	Sets the error topics Kafka topic name		_errors	false	false
kafka.input.topics	Sets the input Kafka topic name		_raw	false	false
kafka.output.topics	Sets the output Kafka topic name		_records	false	false
avro.input.schema	the avro schema definition		null	false	false
avro.output.schema	the avro schema definition for the output serialization		null	false	false
kafka.input.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.output.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
kafka.error.topics.serializer	No Description Provided.	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as	com.hurence.logisland.serializer.JsonSerializer	false	false

Extra informations

No additional information is provided

LocalFileStructuredStreamProviderService

Provide a way to read a local file as input in StructuredStream streams

Class

com.hurence.logisland.stream.spark.structured.provider.LocalFileStructuredStreamProviderService

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 10: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
local.input.path	the location of the directory of files to be loaded. All files inside the directory will be taked as input		null	false	false
max.files.per.trigger	maximum number of new files to be considered in every trigger (default: no max)		null	false	false
latest.first	whether to processs the latest new files first, useful when there is a large backlog of files (default: false)		null	false	false
filename.only	whether to check new files based on only the filename instead of on the full path (default: false). With this set to *true*, the following files would be considered as the same file, because their filenames, "dataset.txt", are the same: "file:///dataset.txt" "s3://a/dataset.txt" "s3n://a/b/dataset.txt" "s3a://a/b/c/dataset.txt"		null	false	false

Extra informations

No additional information is provided

MQTTStructuredStreamProviderService

Provide a ways to use Mqtt a input or output in StructuredStream streams

Class

com.hurence.logisland.stream.spark.structured.provider.MQTTStructuredStreamProviderService

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 11: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mqtt.broker.url	brokerUrl A url MqttClient connects to. Set this or path as the url of the Mqtt Server. e.g. tcp://localhost:1883		tcp://localhost:1883	false	false
mqtt.clean.session	cleanSession Setting it true starts a clean session, removes all checkpointed messages by a previous run of this source. This is set to false by default.		true	false	false
mqtt.client.id	clientID this client is associated. Provide the same value to recover a stopped client.		null	false	false
mqtt.connection.timeout	connectionTimeout Sets the connection timeout, a value of 0 is interpreted as wait until client connects. See MqttConnectOptions.setConnectionTimeout for more information		5000	false	false
mqtt.keep.alive	keepAlive Same as MqttConnectOptions.setKeepAliveInterval.		5000	false	false
mqtt.password	password Sets the password to use for the connection		null	false	false
mqtt.persistence	persistence By default it is used for storing incoming messages on disk. If memory is provided as value for this option, then recovery on restart is not supported.		memory	false	false
mqtt.version	mqttVersion Same as MqttConnectOptions.setMqttVersion		5000	false	false
mqtt.username	username Sets the user name to use for the connection to Mqtt Server. Do not set it, if server does not need this. Setting it empty will lead to errors.		null	false	false
mqtt.qos	QoS The maximum quality of service to subscribe each topic at.Messages published at a lower quality of service will be received at the published QoS.Messages published at a higher quality of service will be received using the QoS specified on the subscribe		0	false	false
mqtt.topic	Topic MqttClient subscribes to.		null	false	false

Extra informations

No additional information is provided

RateStructuredStreamProviderService

Generates data at the specified number of rows per second, each output row contains a timestamp and value. Where timestamp is a Timestamp type containing the time of message dispatch, and value is of Long type containing the message count, starting from 0 as the first row. This source is intended for testing and benchmarking. Used in StructuredStream streams.

Class

com.hurence.logisland.stream.spark.structured.provider.RateStructuredStreamProviderService

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 12: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
local.file.input.path	the location of the file to be loaded		null	false	false
local.file.output.path	the location of the file to be writen		null	false	false
has.csv.header	Is this a csv file with the first line as a header		true	false	false
csv.delimiter	the delimiter		,	false	false

Extra informations

No additional information is provided

RemoteApiStreamProcessingEngine

No description provided.

Class

com.hurence.logisland.engine.spark.RemoteApiStreamProcessingEngine

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 13: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
spark.app.name	The application name		logisland	false	false
spark.master	The url to Spark Master		local[2]	false	false
spark.monitoring.driver.port	The port for exposing monitoring metrics		null	false	false
spark.yarn.deploy-mode	The yarn deploy mode		null	false	false
spark.yarn.queue	The name of the YARN queue		default	false	false
spark.driver.memory	The memory size for Spark driver		512m	false	false
spark.executor.memory	The memory size for Spark executors		1g	false	false
spark.driver.cores	The number of cores for Spark driver		4	false	false
spark.executor.cores	The number of cores for Spark driver		1	false	false
spark.executor.instances	The number of instances for Spark app		null	false	false
spark.serializer	Class to use for serializing objects that will be sent over the network or need to be cached in serialized form		org.apache.spark.serializer.KryoSerializer	false	false
spark.streaming.blockInterval	Interval at which data received by Spark Streaming receivers is chunked into blocks of data before storing them in Spark. Minimum recommended - 50 ms		350	false	false
spark.streaming.kafka.maxRatePerPartition	Maximum rate (number of records per second) at which data will be read from each Kafka partition		5000	false	false
spark.streaming.batchDuration	No Description Provided.		2000	false	false
spark.streaming.backpressure.enabled	This enables the Spark Streaming to control the receiving rate based on the current batch scheduling delays and processing times so that the system receives only as fast as the system can process.		false	false	false
spark.streaming.unpersist	Force RDDs generated and persisted by Spark Streaming to be automatically unpersisted from Spark's memory. The raw input data received by Spark Streaming is also automatically cleared. Setting this to false will allow the raw data and persisted RDDs to be accessible outside the streaming application as they will not be cleared automatically. But it comes at the cost of higher memory usage in Spark.		false	false	false
spark.ui.port	No Description Provided.		4050	false	false
spark.streaming.kafka.maxRetries	No Description Provided.		-1	false	false
spark.streaming.receiver.maxRate	Maximum rate (number of records per second) at which data will be read from each Kafka partition		3	false	false

Continued on next page

Table 13 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
spark.streaming.ui.retainedBatches	How many batches the Spark Streaming UI and status APIs remember before garbage collecting.		200	false	false
spark.streaming.receiver.writeAheadLog.enable	Enable write ahead logs for receivers. All the input data received through receivers will be saved to write ahead logs that will allow it to be recovered after driver failures.		false	false	false
spark.yarn.maxAppAttempts	Because Spark driver and Application Master share a single JVM, any error in Spark driver stops our long-running job. Fortunately it is possible to configure maximum number of attempts that will be made to re-run the application. It is reasonable to set higher value than default 2 (derived from YARN cluster property yarn.resourcemanager.am.max-attempts). 4 works quite well, higher value may cause unnecessary restarts even if the reason of the failure is permanent.		4	false	false
spark.yarn.am.attemptFailuresValidityInterval	If the application runs for days or weeks without restart or redeployment on highly utilized cluster, 4 attempts could be exhausted in few hours. To avoid this situation, the attempt counter should be reset on every hour of so.		1h	false	false
spark.yarn.max.executor.failures	Maximum number of executor failures before the application fails. By default it is max(2 * num executors, 3), well suited for batch jobs but not for long-running jobs. The property comes with corresponding validity interval which also should be set.8 * num_executors		20	false	false
spark.yarn.executor.failuresValidityInterval	If the application runs for days or weeks without restart or redeployment on highly utilized cluster, x attempts could be exhausted in few hours. To avoid this situation, the attempt counter should be reset on every hour of so.		1h	false	false
spark.task.maxFailures	For long-running jobs you could also consider to boost maximum number of task failures before giving up the job. By default tasks will be retried 4 times and then job fails.		8	false	false
spark.memory.fraction	expresses the size of M as a fraction of the (JVM heap space - 300MB) (default 0.75). The rest of the space (25%) is reserved for user data structures, internal metadata in Spark, and safeguarding against OOM errors in the case of sparse and unusually large records.		0.6	false	false

Continued on next page

Table 13 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
spark.memory.storageFraction	Expressed the size of R as a fraction of M (default 0.5). R is the storage space within M where cached blocks immune to being evicted by execution.		0.5	false	false
spark.scheduler.mode	The scheduling mode between jobs submitted to the same SparkContext. Can be set to FAIR to use fair sharing instead of queueing jobs one after another. Useful for multi-user services.	FAIR (fair sharing), FIFO (queueing jobs one after another)	FAIR	false	false
spark.properties.file.path	file path using –properties-file option while submitting spark job		null	false	false
java.library.path	The java library path to use with mesos.		null	false	false
spark.cores.max	The maximum number of total executor core with mesos.		null	false	false
remote.api.baseUrl	The base URL of the remote server providing logisland configuration		null	false	false
remote.api.polling.rate	Remote api polling rate in milliseconds		null	false	false
remote.api.push.rate	Remote api configuration push rate in milliseconds		null	false	false
remote.api.timeout.connect	Remote api connection timeout in milliseconds		10000	false	false
remote.api.auth.user	The basic authentication user for the remote api endpoint.		null	false	false
remote.api.auth.password	The basic authentication password for the remote api endpoint.		null	false	false
remote.api.timeout.socket	Remote api default read/write socket timeout in milliseconds		10000	false	false

Extra informations

No additional information is provided

StructuredStream

No description provided.

Class

com.hurence.logisland.stream.spark.structured.StructuredStream

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 14: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
read.stream.service.provider	The service that gives connection information		null	false	false
read.topics.serializer	The serializer to use	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes), com.hurence.logisland.serializer.KuraProtobufSerializer (serialize events as Kura protocol buffer)	none	false	false
read.topics.key.serializer	The serializer to use	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.KuraProtobufSerializer (serialize events as Kura protocol buffer), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes)	none	false	false
write.stream.service.provider	The service that gives connection information		null	false	false
write.topics.serializer	The serializer to use	com.hurence.logisland.serializer.KryoSerializer (serialize events	none	false	false

Extra informations

No additional information is provided

Engine-vanilla

Find below the list.

AmqpClientPipelineStream

No description provided.

Class

com.hurence.logisland.engine.vanilla.stream.amqp.AmqpClientPipelineStream

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 15: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
connection.host	Connection host name		null	false	false
connection.port	Connection port		5672	false	false
link.credits	Flow control. How many credits for this links. Higher means higher prefetch (pre-buffered number of messages		1024	false	false
connection.auth.user	Connection authenticated user name		null	false	false
connection.auth.password	Connection authenticated password		null	false	false
connection.auth.cert	Connection TLS public certificate (PEM file path)		null	false	false
connection.auth.key	Connection TLS private key (PEM file path)		null	false	false
connection.auth.ca.cert	Connection TLS CA cert (PEM file path)		null	false	false
read.topic	The input path for any topic to be read from			false	false
read.topic.serializer	The serializer to use	com.hurence.logisland.serializer.BsonSerializer (serialize events as bson), com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes), com.hurence.logisland.serializer.KuraProtobufSerializer (serialize events as Kura protocol buffer)	none	false	false
avro.input.schema	The avro schema definition		null	false	false
write.topic	The input path for any topic to be written to			false	false
write.topic.serializer	The serializer to use	com.hurence.logisland.serializer.BsonSerializer (serialize events as bson), com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays),	none	false	false

Extra informations

No additional information is provided

KafkaStreamsPipelineStream

No description provided.

Class

com.hurence.logisland.engine.vanilla.stream.kafka.KafkaStreamsPipelineStream

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 16: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
bootstrap.servers	List of kafka nodes to connect to		null	false	false
read.topics	The input path for any topic to be read from			false	false
avro.input.schema	The avro schema definition		null	false	false
avro.output.schema	The avro schema definition for the output serialization		null	false	false
kafka.manual.offset.reset	What to do when there is no initial offset in Kafka or if the current offset does not exist any more on the server (e.g. because that data has been deleted): earliest: automatically reset the offset to the earliest offset latest: automatically reset the offset to the latest offset none: throw exception to the consumer if no previous offset is found for the consumer's group anything else: throw exception to the consumer.	latest (the offset to the latest offset), earliest (the offset to the earliest offset), none (the latest saved offset)	earliest	false	false
read.topics.serializer	The serializer to use	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), com.hurence.logisland.serializer.BytesArraySerializer (serialize events as byte arrays), com.hurence.logisland.serializer.StringSerializer (serialize events as string), none (send events as bytes), com.hurence.logisland.serializer.KuraProtobufSerializer (serialize events as Kura protocol buffer)	none	false	false
write.topics	The input path for any topic to be written to			false	false
write.topics.serializer	The serializer to use	com.hurence.logisland.serializer.KryoSerializer (serialize events as binary blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.ExtendedJsonSerializer (serialize events as json blocs supporting nested objects/arrays), com.hurence.logisland.serializer.AvroSerializer	none	false	false

Extra informations

No additional information is provided

PlainJavaEngine

No description provided.

Class

com.hurence.logisland.engine.vanilla.PlainJavaEngine

Tags

None.

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 17: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
jvm.heap.min	Minimum memory the JVM should allocate for its heap		null	false	false
jvm.heap.max	Maximum memory the JVM should allocate for its heap		null	false	false

Extra informations

No additional information is provided

Common-processors

Find below the list.

Other-processors

Find below the list.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.1

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 18: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 19: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

"fields": [{ "name": "timestamp", "type": "long" },{ "name": "remoteHost", "type": "string"},{ "name": "record_type", "type": ["null", "string"], "default": null },{ "name": "record_id", "type": ["null", "string"], "default": null },{ "name": "location", "type": ["null", "string"], "default": null },{ "name": "hitType", "type": ["null", "string"], "default": null },{ "name": "eventCategory", "type": ["null", "string"], "default": null },{ "name": "eventAction", "type": ["null", "string"], "default": null },{ "name": "eventLabel", "type": ["null", "string"], "default": null },{ "name": "localPath", "type": ["null", "string"], "default": null },{ "name": "q", "type": ["null", "string"], "default": null },{ "name": "n", "type": ["null", "int"], "default": null },{ "name": "referer", "type": ["null", "string"], "default": null },{ "name": "viewportPixelWidth", "type": ["null", "int"], "default": null },{ "name": "viewportPixelHeight", "type": ["null", "int"], "default": null },{ "name": "screenPixelWidth", "type": ["null", "int"], "default": null },{ "name": "screenPixelHeight", "type": ["null", "int"], "default": null },{ "name": "partyId", "type": ["null", "string"], "default": null },{ "name": "sessionId", "type": ["null", "string"], "default": null },{ "name": "pageViewId", "type": ["null", "string"], "default": null },{ "name": "is_newSession", "type": ["null", "boolean"],"default": null },{ "name": "userAgentString", "type": ["null", "string"], "default": null },{ "name": "pageType", "type": ["null", "string"], "default": null },{ "name": "UserId", "type": ["null", "string"], "default": null },{ "name": "B2Bunit", "type": ["null", "string"], "default": null },{ "name": "pointOfService", "type": ["null", "string"], "default": null },{ "name": "companyID", "type": ["null", "string"], "default": null },{ "name": "GroupCode", "type": ["null", "string"], "default": null },{ "name": "userRoles", "type": ["null", "string"], "default": null },{ "name": "is_PunchOut", "type": ["null", "string"], "default": null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 20: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample

- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 21: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 22: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 23: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 24: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 25: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 26: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web

session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 27: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 28: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 29: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 30: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 31: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 32: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 33: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 34: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{
"conn": {
"id.resp_p": 9092,
"resp_pkts": 0,
"resp_ip_bytes": 0,
"local_orig": true,
"orig_ip_bytes": 0,
"orig_pkts": 0,
"missed_bytes": 0,
"history": "Cc",
"tunnel_parents": [],
"id.orig_p": 56762,
"local_resp": true,
"uid": "Ct3Ms01I3Yc6pmMZx7",
"conn_state": "OTH",
"id.orig_h": "172.17.0.2",
"proto": "tcp",
"id.resp_h": "172.17.0.3",
"ts": 1487596886.953917
}
}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"
"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"
"record_time": 1487597792782
"record_type": "conn"
"id_resp_p": 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 35: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 36: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 37: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 38: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 39: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 40: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 41: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

---

setSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the setSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.setSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 42: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the setSourceOfTraffic processor needs to have access to an

Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 43: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 44: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 45: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 46: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 47: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 48: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 49: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 50: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 51: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 52: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 53: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 54: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 55: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 56: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 57: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 58: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 59: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 60: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 61: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 62: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 63: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 64: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 65: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 66: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 67: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 68: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 69: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 70: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 71: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 72: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 73: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 74: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 75: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 76: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 77: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 78: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 79: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 80: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 81: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 82: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 83: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 84: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 85: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 86: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 87: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 88: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 89: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 90: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 91: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 92: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 93: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 94: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 95: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 96: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 97: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 98: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 99: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 100: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 101: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 102: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 103: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 104: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 105: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 106: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 107: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 108: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 109: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 110: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 111: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 112: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 113: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 114: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 115: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 116: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 117: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 118: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 119: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 120: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 121: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 122: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 123: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 124: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 125: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 126: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 127: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 128: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 129: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 130: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 131: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 132: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 133: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 134: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 135: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 136: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 137: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 138: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917
}
}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"
"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"
"record_time": 1487597792782
"record_type": "conn"
"id_resp_p": 9092
"resp_pkts": 0
"resp_ip_bytes": 0
"local_orig": true
"orig_ip_bytes": 0
"orig_pkts": 0
"missed_bytes": 0
"history": "Cc"
"tunnel_parents": []
"id_orig_p": 56762
"local_resp": true
"uid": "Ct3Ms01I3Yc6pmMZx7"
"conn_state": "OTH"
"id_orig_h": "172.17.0.2"
"proto": "tcp"
"id_resp_h": "172.17.0.3"
"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 139: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 140: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 141: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 142: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 143: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 144: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 145: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 146: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 147: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 148: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 149: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 150: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 151: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 152: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 153: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 154: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 155: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 156: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 157: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 158: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 159: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 160: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 161: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 162: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 163: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 164: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 165: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 166: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 167: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 168: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 169: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 170: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 171: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 172: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 173: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 174: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 175: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 176: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 177: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 178: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 179: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 180: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 181: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 182: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 183: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 184: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 185: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 186: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 187: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 188: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 189: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 190: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 191: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 192: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 193: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 194: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 195: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 196: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 197: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 198: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 199: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 200: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 201: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 202: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 203: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 204: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 205: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 206: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 207: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 208: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 209: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 210: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 211: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 212: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 213: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 214: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 215: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 216: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 217: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 218: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 219: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 220: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 221: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 222: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 223: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 224: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 225: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 226: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 227: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 228: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 229: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 230: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 231: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 232: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 233: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 234: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 235: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 236: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 237: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 238: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 239: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 240: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 241: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 242: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 243: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 244: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 245: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 246: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 247: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 248: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 249: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 250: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 251: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 252: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 253: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 254: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 255: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 256: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 257: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 258: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 259: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 260: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 261: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 262: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 263: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 264: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 265: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 266: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 267: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 268: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 269: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 270: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 271: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 272: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 273: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 274: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 275: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 276: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 277: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 278: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 279: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 280: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 281: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimumPointsToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 282: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 283: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 284: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 285: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 286: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 287: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 288: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 289: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 290: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 291: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 292: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 293: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 294: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 295: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 296: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 297: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 298: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 299: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 300: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 301: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 302: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 303: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 304: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 305: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 306: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 307: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 308: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 309: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 310: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 311: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 312: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 313: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917
}
}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"
"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"
"record_time": 1487597792782
"record_type": "conn"
"id_resp_p": 9092
"resp_pkts": 0
"resp_ip_bytes": 0
"local_orig": true
"orig_ip_bytes": 0
"orig_pkts": 0
"missed_bytes": 0
"history": "Cc"
"tunnel_parents": []
"id_orig_p": 56762
"local_resp": true
"uid": "Ct3Ms01I3Yc6pmMZx7"
"conn_state": "OTH"
"id_orig_h": "172.17.0.2"
"proto": "tcp"
"id_resp_h": "172.17.0.3"
"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 314: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 315: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 316: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 317: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 318: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 319: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 320: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 321: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 322: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 323: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 324: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 325: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 326: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 327: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 328: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 329: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 330: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 331: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 332: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 333: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 334: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 335: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 336: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 337: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 338: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 339: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 340: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 341: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 342: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 343: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 344: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 345: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 346: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 347: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 348: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 349: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 350: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 351: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 352: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 353: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 354: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 355: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 356: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 357: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 358: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 359: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 360: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 361: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 362: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 363: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 364: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 365: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 366: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 367: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 368: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 369: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 370: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 371: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 372: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 373: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 374: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 375: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 376: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 377: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 378: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 379: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 380: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 381: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 382: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 383: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 384: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 385: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 386: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 387: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 388: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 389: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 390: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 391: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 392: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 393: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 394: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 395: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 396: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 397: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 398: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 399: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 400: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 401: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 402: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 403: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 404: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 405: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 406: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 407: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 408: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 409: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 410: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 411: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 412: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 413: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 414: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 415: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 416: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 417: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 418: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 419: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 420: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 421: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 422: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 423: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 424: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 425: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 426: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 427: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 428: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 429: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 430: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 431: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 432: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 433: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 434: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 435: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 436: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 437: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 438: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 439: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 440: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 441: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 442: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 443: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 444: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 445: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 446: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 447: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 448: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 449: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 450: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 451: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 452: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 453: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 454: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 455: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 456: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 457: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 458: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 459: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 460: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 461: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 462: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 463: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 464: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 465: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 466: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 467: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 468: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 469: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 470: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 471: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 472: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 473: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 474: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 475: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 476: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 477: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 478: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 479: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 480: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 481: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 482: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 483: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 484: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 485: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 486: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 487: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 488: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 489: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 490: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 491: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 492: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 493: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 494: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 495: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 496: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 497: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 498: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 499: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 500: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 501: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 502: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 503: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 504: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 505: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	The name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	The name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 506: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 507: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 508: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 509: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 510: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 511: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 512: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 513: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 514: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 515: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 516: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 517: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 518: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 519: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 520: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 521: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 522: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 523: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 524: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 525: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 526: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 527: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 528: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 529: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 530: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 531: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to use to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 532: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 533: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 534: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 535: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 536: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 537: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 538: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 539: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 540: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 541: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 542: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 543: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 544: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 545: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 546: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 547: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 548: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 549: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 550: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 551: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 552: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 553: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 554: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 555: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 556: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 557: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 558: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 559: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 560: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 561: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 562: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 563: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 564: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 565: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 566: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 567: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.3.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 568: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.3.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 569: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.3.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 570: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.3.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 571: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.3.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 572: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.3.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 573: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.3.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 574: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.3.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 575: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.3.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 576: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.3.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 577: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.3.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 578: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 579: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.3.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 580: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.3.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 581: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum amount to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.3.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 582: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.3.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 583: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.3.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 584: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 585: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.3.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 586: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 587: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.3.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 588: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.3.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 589: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.3.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 590: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.3.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 591: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 592: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 593: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 594: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 595: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 596: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 597: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 598: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 599: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 600: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 601: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 602: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 603: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 604: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 605: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 606: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 607: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 608: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 609: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 610: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 611: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 612: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 613: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917
}
}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"
"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"
"record_time": 1487597792782
"record_type": "conn"
"id_resp_p": 9092
"resp_pkts": 0
"resp_ip_bytes": 0
"local_orig": true
"orig_ip_bytes": 0
"orig_pkts": 0
"missed_bytes": 0
"history": "Cc"
"tunnel_parents": []
"id_orig_p": 56762
"local_resp": true
"uid": "Ct3Ms01I3Yc6pmMZx7"
"conn_state": "OTH"
"id_orig_h": "172.17.0.2"
"proto": "tcp"
"id_resp_h": "172.17.0.3"
"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 614: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 615: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 616: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 617: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 618: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 619: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 620: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 621: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 622: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 623: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 624: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 625: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 626: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 627: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 628: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 629: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 630: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 631: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 632: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 633: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 634: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 635: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 636: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 637: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 638: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 639: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 640: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 641: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 642: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 643: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 644: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 645: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 646: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 647: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 648: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 649: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 650: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 651: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 652: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 653: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 654: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 655: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 656: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum points to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 657: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 658: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 659: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 660: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 661: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 662: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 663: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 664: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 665: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 666: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 667: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 668: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 669: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 670: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 671: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 672: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 673: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 674: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 675: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 676: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 677: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 678: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 679: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 680: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 681: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 682: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 683: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 684: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 685: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 686: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 687: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 688: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 689: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 690: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 691: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 692: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 693: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 694: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 695: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 696: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 697: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 698: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 699: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 700: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 701: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 702: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 703: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 704: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 705: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 706: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum amount to predict		100	false	false
min_zscore_percentile	min Zscore percentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 707: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 708: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 709: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 710: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 711: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 712: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 713: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 714: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 715: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 716: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 717: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 718: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 719: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 720: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 721: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 722: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 723: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 724: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 725: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 726: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 727: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 728: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 729: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 730: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 731: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to use to predict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 732: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 733: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 734: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 735: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 736: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 737: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 738: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 739: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 740: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 741: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 742: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 743: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 744: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 745: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 746: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 747: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 748: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 749: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 750: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 751: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 752: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 753: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 754: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 755: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 756: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 757: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 758: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 759: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 760: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 761: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 762: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 763: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 764: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 765: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 766: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 767: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 768: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 769: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 770: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 771: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 772: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 773: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 774: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 775: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 776: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 777: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 778: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 779: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 780: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 781: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 782: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minimum number of points to start computing zscore		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 783: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 784: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 785: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 786: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 787: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 788: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 789: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 790: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 791: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 792: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 793: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 794: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 795: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 796: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 797: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 798: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 799: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 800: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 801: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 802: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 803: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 804: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 805: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 806: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 807: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 808: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 809: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 810: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 811: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 812: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 813: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 814: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 815: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 816: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 817: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 818: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 819: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 820: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 821: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 822: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 823: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 824: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 825: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 826: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 827: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 828: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 829: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 830: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 831: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 832: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 833: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 834: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 835: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 836: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 837: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 838: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 839: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 840: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 841: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 842: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 843: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 844: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 845: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.suffix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 846: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 847: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 848: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 849: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 850: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 851: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 852: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 853: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 854: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 855: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 856: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 857: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 858: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 859: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 860: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 861: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 862: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 863: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 864: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 865: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 866: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 867: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 868: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 869: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 870: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 871: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 872: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 873: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 874: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 875: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 876: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 877: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 878: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 879: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 880: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 881: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 882: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 883: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 884: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 885: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 886: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 887: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 888: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 889: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 890: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 891: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 892: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 893: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 894: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 895: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 896: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 897: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 898: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 899: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 900: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 901: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 902: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 903: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 904: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 905: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 906: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 907: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 908: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 909: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 910: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 911: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 912: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 913: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 914: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 915: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 916: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 917: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 918: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 919: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 920: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 921: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 922: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 923: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 924: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 925: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 926: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 927: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 928: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 929: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 930: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 931: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 932: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 933: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 934: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 935: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 936: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 937: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 938: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 939: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 940: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 941: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 942: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 943: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 944: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 945: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 946: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 947: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 948: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 949: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 950: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 951: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 952: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 953: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 954: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 955: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 956: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 957: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 958: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 959: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 960: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 961: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 962: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 963: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 964: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 965: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 966: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 967: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 968: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 969: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 970: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 971: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 972: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 973: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 974: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 975: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 976: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 977: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 978: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 979: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 980: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 981: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 982: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 983: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 984: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 985: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 986: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 987: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 988: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 989: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 990: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 991: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 992: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 993: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 994: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 995: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 996: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 997: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 998: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 999: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1000: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1001: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1002: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1003: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1004: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1005: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1006: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1007: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1008: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1009: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1010: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1011: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1012: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1013: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1014: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1015: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1016: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1017: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1018: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1019: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1020: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1021: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1022: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1023: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1024: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1025: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1026: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1027: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1028: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1029: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1030: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1031: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1032: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1033: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1034: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1035: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1036: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1037: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1038: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1039: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1040: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1041: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1042: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1043: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1044: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1045: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1046: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1047: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1048: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1049: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1050: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1051: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1052: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1053: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1054: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1055: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1056: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1057: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1058: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1059: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1060: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1061: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1062: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1063: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1064: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1065: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1066: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1067: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1068: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1069: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1070: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1071: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1072: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1073: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1074: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1075: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1076: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1077: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1078: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1079: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1080: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1081: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webAnalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1082: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1083: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1084: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1085: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1086: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1087: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1088: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1089: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1090: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1091: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webAnalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1092: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1093: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
fields to decode	a default value	Decode one or more fields from the record		null	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1094: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1095: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1095 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1095 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1095 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1096: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1097: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1098: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1099: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1100: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1101: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1102: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1103: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1104: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1105: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1106: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1107: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1108: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1109: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1110: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1111: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1112: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1113: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1114: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1115: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1116: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1117: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1118: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1119: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1120: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1121: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1121 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1121 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1121 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field	The name of the field where index name to store record will be stored		es_index	false	false
record.es.type.field	The name of the field where type name to store record will be stored		es_type	false	false
number.of.future.sessions	The number of sessions to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1122: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The name of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1123: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1124: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1125: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1126: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1127: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1128: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1129: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1130: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1131: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1132: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1133: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1134: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1135: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1136: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1137: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1138: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1139: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1140: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1141: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1142: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1143: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1144: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1145: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1146: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1147: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1147 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1147 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1147 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1148: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1149: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1150: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1151: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1152: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1153: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1154: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1155: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1156: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1157: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1158: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1159: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1160: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1161: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1162: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1163: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1164: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1165: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1166: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1167: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1168: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1169: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1170: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1171: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1172: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1173: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1173 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.name.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1173 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1173 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field	The name of the field where index name to store record will be stored		es_index	false	false
record.es.type.field	The name of the field where type name to store record will be stored		es_type	false	false
number.of.future.sessions	The number of sessions to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1174: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1175: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1176: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1177: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1178: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1179: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1180: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1181: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1182: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1183: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1184: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1185: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1186: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1187: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1188: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1189: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1190: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1191: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1192: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1193: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1194: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1195: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1196: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1197: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1198: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1199: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1199 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1199 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1199 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1200: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1201: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1202: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1203: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1204: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1205: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1206: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1207: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1208: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1209: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1210: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1211: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1212: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1213: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1214: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1215: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1216: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1217: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1218: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1219: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1220: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1221: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1222: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1223: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1224: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1225: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1225 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1225 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1225 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1226: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1227: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1228: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1229: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1230: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1231: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1232: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1233: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1234: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1235: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1236: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1237: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1238: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1239: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1240: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1241: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1242: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1243: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1244: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1245: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1246: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1247: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1248: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1249: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1250: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1251: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1251 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1251 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1251 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1252: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1253: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1254: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1255: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1256: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1257: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1258: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1259: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1260: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1261: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1262: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1263: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1264: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1265: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1266: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1267: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1268: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1269: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1270: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1271: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1272: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1273: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1274: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1275: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1276: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1277: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1277 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1277 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1277 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.wait	The number of session to wait for when searching session of last events		1	false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1278: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false

Continued on next page

Table 1278 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
firstEventDateTime.name	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.name	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.name	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.name	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1278 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1278 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field	The name of the field where index name to store record will be stored		es_index	false	false
record.es.type.field	The name of the field where type name to store record will be stored		es_type	false	false
number.of.future.session	The number of session from now to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1279: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1280: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1281: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1282: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1283: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1284: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1285: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1286: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1287: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1288: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1289: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1290: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1291: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1292: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1293: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1294: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1295: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1296: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1297: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1298: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1299: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1300: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1301: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1302: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1303: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1304: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1304 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1304 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1304 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field.name	The field where index name to store record will be stored		es_index	false	false
record.es.type.field.name	The field where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session from past to look for when searching session of last events		1	false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1305: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false

Continued on next page

Table 1305 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
firstEventDateTime.name	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.name	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.name	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.name	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1305 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	Dynamic
es.index.suffix.timezone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1305 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1306: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.suffix	Suffix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The name of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1307: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1308: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1309: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1310: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1311: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1312: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1313: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1314: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1315: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1316: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1317: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1318: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1319: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1320: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1321: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1322: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1323: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1324: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1325: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1326: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1327: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1328: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1329: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1330: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1331: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
es.index.suffix.timezone	The timezone to use to aprse timestamp into string date (for index		null	false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1332: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1332 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1332 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
es.index.suffix.timezone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1332 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1333: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The name of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1334: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1335: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1336: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1337: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1338: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1339: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1340: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1341: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1342: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1343: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1344: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1345: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1346: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1347: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1348: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1349: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1350: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1351: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1352: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1353: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1354: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1355: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1356: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1357: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1358: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
es.index.suffix.timezone	The timezone to use to aprse timestamp into string date (for index		null	false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1359: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1359 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1359 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1359 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field	The name of the field where index name to store record will be stored		es_index	false	false
record.es.type.field	The name of the field where type name to store record will be stored		es_type	false	false
number.of.future.sessions	The number of sessions to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1360: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The name of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1361: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1362: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1363: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1364: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1365: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1366: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1367: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1368: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1369: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1370: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1371: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1372: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1373: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1374: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1375: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1376: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1377: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1378: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1379: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1380: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1381: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1382: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1383: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1384: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1385: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
es.index.suffix.timezone	The timezone to use to aprse timestamp into string date (for index		null	false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1386: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1386 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1386 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1386 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1387: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The name of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution

occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1388: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1389: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1390: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1391: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1392: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1393: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1394: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1395: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1396: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1397: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1398: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1399: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1400: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1401: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1402: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1403: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1404: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

```
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1405: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1406: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1407: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1408: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1409: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1410: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1411: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1412: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
es.index.suffix.timezone	The timezone to use to aprse timestamp into string date (for index		null	false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1413: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false

Continued on next page

Table 1413 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.output.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1413 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1413 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

IncrementalWebSessionOld

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSessionOld

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1414: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1415: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1416: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1417: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1418: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1419: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1420: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1421: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1422: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1423: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1424: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1425: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1426: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1427: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		default	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1428: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
```

(continues on next page)

(continued from previous page)

```
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1429: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1430: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1431: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1432: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

"conn": {

"id.resp_p": 9092,

"resp_pkts": 0,

"resp_ip_bytes": 0,

"local_orig": true,

"orig_ip_bytes": 0,

"orig_pkts": 0,

"missed_bytes": 0,

"history": "Cc",

"tunnel_parents": [],

"id.orig_p": 56762,

"local_resp": true,

"uid": "Ct3Ms01I3Yc6pmMZx7",

"conn_state": "OTH",

"id.orig_h": "172.17.0.2",

"proto": "tcp",

“id.resp_h”: “172.17.0.3”,

“ts”: 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

“@timestamp”: “2017-02-20T13:36:32Z”

“record_id”: “6361f80a-c5c9-4a16-9045-4bb51736333d”

“record_time”: 1487597792782

“record_type”: “conn”

“id_resp_p”: 9092

“resp_pkts”: 0

“resp_ip_bytes”: 0

“local_orig”: true

“orig_ip_bytes”: 0

“orig_pkts”: 0

“missed_bytes”: 0

“history”: “Cc”

“tunnel_parents”: []

“id_orig_p”: 56762

“local_resp”: true

“uid”: “Ct3Ms01I3Yc6pmMZx7”

“conn_state”: “OTH”

“id_orig_h”: “172.17.0.2”

“proto”: “tcp”

“id_resp_h”: “172.17.0.3”

“ts”: 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1433: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

“uid”: “Ct3Ms01I3Yc6pmMZx7”,

“conn_state”: “OTH”,

“id.orig_h”: “172.17.0.2”,

“proto”: “tcp”,

“id.resp_h”: “172.17.0.3”,

"ts": 1487596886.953917

}

}

It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1434: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1435: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1436: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1437: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1438: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1439: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1440: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.out.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source.of.traffic.prefix	prefix for the source of the traffic related			false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1441: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.out.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false

Continued on next page

Table 1441 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
lastEventDateTime.name	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.name	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1441 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1441 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false
processing.mode	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.	FAST, MODERATE, SLOW	FAST	false	false
es.refresh.wait.time	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.		100000	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

IncrementalWebSessionOld

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web

session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSessionOld

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1442: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1443: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1444: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1445: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1446: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1447: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

FetchHBaseRow

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.FetchHBaseRow

Tags

hbase, scan, fetch, get, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1448: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to fetch from.		null	false	**true**
row.identifier.field	The field containing the identifier of the row to fetch.		null	false	**true**
columns.field	The field containing an optional comma-separated list of “”<colFamily>:<colQualifier>”” pairs to fetch. To return all columns for a given family, leave off the qualifier such as “”<colFamily1>,<colFamily2>””.		null	false	**true**
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
table.name.default	The table to use if table name field is not set		null	false	false

Extra informations

Fetches a row from an HBase table. The Destination property controls whether the cells are added as flow file attributes, or the row is written to the flow file content as JSON. This processor may be used to fetch a fixed row on a interval by specifying the table and row id directly in the processor, or it may be used to dynamically fetch rows by referencing the table and row id from incoming flow files.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1449: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

PutHBaseCell

Adds the Contents of a Record to HBase as the value of a single cell

Module

com.hurence.logisland:logisland-processor-hbase:1.4.0

Class

com.hurence.logisland.processor.hbase.PutHBaseCell

Tags

hadoop, hbase

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1450: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hbase.client.service	The instance of the Controller Service to use for accessing HBase.		null	false	false
table.name.field	The field containing the name of the HBase Table to put data into		null	false	**true**
row.identifier.field	Specifies field containing the Row ID to use when inserting data into HBase		null	false	**true**
row.identifier.encoding.strategy	Specifies the data type of Row ID used when inserting data into HBase. The default behavior is to convert the row id to a UTF-8 byte array. Choosing Binary will convert a binary formatted string to the correct byte[] representation. The Binary option should be used if you are using Binary row keys in HBase	String (Stores the value of row id as a UTF-8 String.), Binary (Stores the value of the rows id as a binary byte array. It expects that the row id is a binary formatted string.)	String	false	false
column.family.field	The field containing the Column Family to use when inserting data into HBase		null	false	**true**
column.qualifier.field	The field containing the Column Qualifier to use when inserting data into HBase		null	false	**true**
batch.size	The maximum number of Records to process in a single execution. The Records will be grouped by table, and a single Put per table will be performed.		25	false	false
record.schema	the avro schema definition for the Avro serialization		null	false	false
record.serializer	the serializer needed to i/o the record in the HBase row	com.hurence.logisland.serializer.KryoSerializer (serialize events as json blocs), com.hurence.logisland.serializer.JsonSerializer (serialize events as json blocs), com.hurence.logisland.serializer.AvroSerializer (serialize events as avro blocs), none (send events as bytes)	com.hurence.logisland.serializer.KryoSerializer	false	false
table.name.default	The table table to use if table name field is not set		null	false	false
column.family.default	The column family to use if column family field is not set		null	false	false
column.qualifier.default	The column qualifier to use if column qualifier field is not set		null	false	false

Extra informations

Adds the Contents of a Record to HBase as the value of a single cell.

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1451: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1452: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field:

Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1453: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1454: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1455: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		_doc	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false
cache.service	The instance of the Cache Service to use (optional).		null	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1456: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1457: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1458: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1459: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1460: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

```
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1461: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{
“conn”: {
“id.resp_p”: 9092,
“resp_pkts”: 0,
“resp_ip_bytes”: 0,
“local_orig”: true,
“orig_ip_bytes”: 0,
“orig_pkts”: 0,
“missed_bytes”: 0,
“history”: “Cc”,
“tunnel_parents”: [],

```
            "id.orig_p": 56762,
            "local_resp": true,
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1462: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors

- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1463: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1464: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1465: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1466: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1467: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1468: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.out.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source.of.traffic.prefix	prefix for the source of the traffic related			false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1469: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.out.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false

Continued on next page

Table 1469 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
lastEventDateTime.name	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.name	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1469 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1469 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session from now to look for when searching session of last events		1	false	false
processing.mode	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.	FAST, MODERATE, SLOW	FAST	false	false
es.refresh.wait.time	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.		100000	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

IncrementalWebSessionOld

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web

session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSessionOld

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1470: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1471: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1472: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1473: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	Expression Language
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1474: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1475: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.

- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1476: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1477: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1478: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”,

“int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1479: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1480: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1481: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		_doc	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false
cache.service	The instance of the Cache Service to use (optional).		null	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1482: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1483: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1484: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1485: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1486: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

```
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1487: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{
“conn”: {
“id.resp_p”: 9092,
“resp_pkts”: 0,
“resp_ip_bytes”: 0,
“local_orig”: true,
“orig_ip_bytes”: 0,
“orig_pkts”: 0,
“missed_bytes”: 0,
“history”: “Cc”,
“tunnel_parents”: [],

```
            "id.orig_p": 56762,
            "local_resp": true,
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1488: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors

- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1489: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1490: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1491: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1492: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1493: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1494: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source.of.traffic.suffix	suffix for the source of the traffic related			false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1495: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.out.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false

Continued on next page

Table 1495 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
lastEventDateTime.name	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.name	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1495 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1495 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false
processing.mode	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.	FAST, MODERATE, SLOW	FAST	false	false
es.refresh.wait.time	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.		100000	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

IncrementalWebSessionOld

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web

session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSessionOld

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1496: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1497: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Time out validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1498: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1499: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1500: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1501: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.

- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1502: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1503: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1504: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”,

“int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1505: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1506: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1507: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		_doc	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false
cache.service	The instance of the Cache Service to use (optional).		null	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1508: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1509: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1510: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1511: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1512: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

```
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1513: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{
“conn”: {
“id.resp_p”: 9092,
“resp_pkts”: 0,
“resp_ip_bytes”: 0,
“local_orig”: true,
“orig_ip_bytes”: 0,
“orig_pkts”: 0,
“missed_bytes”: 0,
“history”: “Cc”,
“tunnel_parents”: [],

```
            "id.orig_p": 56762,
            "local_resp": true,
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1514: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors

- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1515: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1516: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1517: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1518: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

ParseUserAgent

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

Module

com.hurence.logisland:logisland-processor-useragent:1.4.0

Class

com.hurence.logisland.processor.useragent.ParseUserAgent

Tags

User-Agent, clickstream, DMP

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1519: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
cache.enabled	Enable caching. Caching to avoid to redo the same computation for many identical User-Agent strings.		true	false	false
cache.size	Set the size of the cache.		1000	false	false
useragent.field	Must contain the name of the field that contains the User-Agent value in the incoming record.		null	false	false
useragent.keep	Defines if the field that contained the User-Agent must be kept or not in the resulting records.		true	false	false
confidence.enabled	Enable confidence reporting. Each field will report a confidence attribute with a value comprised between 0 and 10000.		false	false	false
ambiguity.enabled	Enable ambiguity reporting. Reports a count of ambiguities.		false	false	false
fields	Defines the fields to be returned.		DeviceClass, DeviceName, DeviceBrand, DeviceCpu, DeviceFirmwareVersion, DeviceVersion, OperatingSystemClass, OperatingSystemName, OperatingSystemVersion, OperatingSystemNameVersion, OperatingSystemVersionBuild, LayoutEngineClass, LayoutEngineName, LayoutEngineVersion, LayoutEngineVersionMajor, LayoutEngineNameVer-	false	false

Extra informations

The user-agent processor allows to decompose User-Agent value from an HTTP header into several attributes of interest. There is no standard format for User-Agent strings, hence it is not easily possible to use regexp to handle them. This processor rely on the YAUAA library to do the heavy work.

CalculWebSession

This processor creates web-sessions based on incoming web-events. Firstly, web-events are grouped by their session identifier and processed in chronological order. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.CalculWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1520: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.out.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source.of.traffic.prefix	prefix for the source of the traffic related			false	false

Extra informations

IncrementalWebSession

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1521: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.prefix	Prefix of the indices containing the web session documents.		null	false	false
es.session.index.suffix.date	Suffix to add to prefix for web session indices. It should be valid date format [yyyy.MM].		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.index.suffix.date	Suffix to add to prefix for web event indices. It should be valid date format [yyyy.MM].		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSinglePageVisit.out.field	the name of the field stating whether the session is single page visit or not => will override default value if set		is_single_page_visit	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false

Continued on next page

Table 1521 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
lastEventDateTime.name	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.name	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic_	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	The name of the cache service to use.		null	false	false

Continued on next page

Table 1521 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	El
es.index.suffix.time.zone	The timezone to use to aprse timestamp into string date (for index names). See es.event.index.suffix.date and es.session.index.suffix.date. By default the system timezone is used. Supported by current system is : [Asia/Aden, America/Cuiaba, Etc/GMT+9, Etc/GMT+8, Africa/Nairobi, America/Marigot, Asia/Aqtau, Pacific/Kwajalein, America/El_Salvador, Asia/Pontianak, Africa/Cairo, Pacific/Pago_Pago, Africa/Mbabane, Asia/Kuching, Pacific/Honolulu, Pacific/Rarotonga, America/Guatemala, Australia/Hobart, Europe/London, America/Belize, America/Panama, Asia/Chungking, America/Managua, America/Indiana/Petersburg, Asia/Yerevan, Europe/Brussels, GMT, Europe/Warsaw, America/Chicago, Asia/Kashgar, Chile/Continental, Pacific/Yap, CET, Etc/GMT-1, Etc/GMT-0, Europe/Jersey, America/Tegucigalpa, Etc/GMT-5, Europe/Istanbul, America/Eirunepe, Etc/GMT-4, America/Miquelon, Etc/GMT-3, Europe/Luxembourg, Etc/GMT-2, Etc/GMT-9, America/Argentina/Catamarca, Etc/GMT-8, Etc/GMT-7, Etc/GMT-6, Europe/Zaporozhye, Canada/Yukon, Canada/Atlantic, Atlantic/St_Helena, Australia/Tasmania, Libya, Europe/Guernsey, America/Grand_Turk, US/Pacific-New, Asia/Samarkand, America/Argentina/Cordoba, Asia/Phnom_Penh, Africa/Kigali, Asia/Almaty, US/Alaska, Asia/Dubai, Europe/Isle_of_Man, America/Araguaina, Cuba, Asia/Novosibirsk, America/Argentina/Salta, Etc/GMT+3, Africa/Tunis, Etc/GMT+2, Etc/GMT+1, Pacific/Fakaofo, Africa/Tripoli, Etc/GMT+0, Israel, Africa/Banjul, Etc/GMT+7, Indian/Comoro, Etc/GMT+6, Etc/GMT+5, Etc/GMT+4, Pacific/Port_Moresby, US/Arizona, Antarctica/Syowa, Indian/Reunion, Pacific/Palau, Europe/Kaliningrad, America/Montevideo, Africa/Windhoek, Asia/Karachi, Africa/Mogadishu, Australia/Perth, Brazil/East, Etc/GMT, Asia/Chita, Pacific/Easter, Antarctica/Davis, Antarctica/McMurdo, Asia/Macao, America/Manaus, Africa/Freetown, Europe/Bucharest, Asia/Tomsk, America/Argentina/Mendoza, Asia/Macau, Europe/Malta, Mexico/BajaSur, Pacific/Tahiti, Africa/Asmera, Europe/Busingen, America/Argentina/Rio_Gallegos,		null	false	false

Table 1521 – continued from previous page

Name	Description	Allowable Values	Default Value	Sensitive	EL
record.es.index.output.field.name	The field name where index name to store record will be stored		es_index	false	false
record.es.type.output.field.name	The field name where type name to store record will be stored		es_type	false	false
number.of.future.session.to.look	The number of session to look for when searching session of last events		1	false	false
processing.mode	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.	FAST, MODERATE, SLOW	FAST	false	false
es.refresh.wait.time	If fastMode is true the processor will not do refresh on es indices which will improve performance but The result may be not exact as we are not sure to query the events up to date.		100000	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

> the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

IncrementalWebSessionOld

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events are

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web

session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.IncrementalWebSessionOld

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1522: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, debug information are logged.		false	false	false
es.session.index.field	Name of the field in the record defining the ES index containing the web session documents.		null	false	false
es.session.type.name	Name of the ES type of web session documents.		null	false	false
es.event.index.prefix	Prefix of the index containing the web event documents.		null	false	false
es.event.type.name	Name of the ES type of web event documents.		null	false	false
es.mapping.event.to.session.index.name	Name of the ES index containing the mapping of web session documents.		null	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.out.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.out.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.out.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.out.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
sessionInactivityDuration.out.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false
session.timeout	session timeout in sec		1800	false	false
eventsCounter.out.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.out.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.out.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
newSessionReason.out.field	the name of the field containing the reason why a new session was created => will override default value if set		reasonForNewSession	false	false
transactionIds.out.field	the name of the field containing all transactionIds => will override default value if set		transactionIds	false	false
source_of_traffic.suffix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false

Extra informations

This processor creates and updates web-sessions based on incoming web-events. Note that both web-sessions and web-events ar

Firstly, web-events are grouped by their session identifier and processed in chronological order. Then each web-session associated to each group is retrieved from elasticsearch. In case none exists yet then a new web session is created based on the first web event. The following fields of the newly created web session are set based on the associated web event: session identifier, first timestamp, first visited page. Secondly, once created, or retrieved, the web session is updated by the remaining web-events. Updates have impacts on fields of the web session such as event counter, last visited page, session duration, ... Before updates are actually applied, checks are performed to detect rules that would trigger the creation of a new session:

the duration between the web session and the web event must not exceed the specified time-out, the web session and the web event must have timestamps within the same day (at midnight a new web session is created), source of traffic (campaign, ...) must be the same on the web session and the web event.

When a breaking rule is detected, a new web session is created with a new session identifier where as remaining web-events still have the original session identifier. The new session identifier is the original session suffixed with the character '#' followed with an incremented counter. This new session identifier is also set on the remaining web-events. Finally when all web events were applied, all web events -potentially modified with a new session identifier- are save in elasticsearch. And web sessions are passed to the next processor.

WebSession information are: - first and last visited page - first and last timestamp of processed event - total number of processed events - the userId - a boolean denoting if the web-session is still active or not - an integer denoting the duration of the web-sessions - optional fields that may be retrieved from the processed events

---

SetSourceOfTraffic

Compute the source of traffic of a web session. Users arrive at a website or application through a variety of sources, including advertising/paying campaigns, search engines, social networks, referring sites or direct access. When analysing user experience on a webshop, it is crucial to collect, process, and report the campaign and traffic-source data. To compute the source of traffic of a web session, the user has to provide the utm_* related properties if available i-e: **utm_source.field**, **utm_medium.field**, **utm_campaign.field**, **utm_content.field**, **utm_term.field**) , the referer (**referer.field** property) and the first visited page of the session (**first.visited.page.field** property). By default the source of traffic information are placed in a flat structure (specified by the **source_of_traffic.prefix** property with a default value of source_of_traffic). To work properly the SetSourceOfTraffic processor needs to have access to an Elasticsearch index containing a list of the most popular search engines and social networks. The ES index (specified by the **es.index** property) should be structured such that the _id of an ES document MUST be the name of the domain. If the domain is a search engine, the related ES doc MUST have a boolean field (default being search_engine) specified by the property **es.search_engine.field** with a value set to true. If the domain is a social network , the related ES doc MUST have a boolean field (default being social_network) specified by the property **es.social_network.field** with a value set to true.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.SetSourceOfTraffic

Tags

session, traffic, source, web, analytics

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1523: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
referer.field	Name of the field containing the referer value in the session		referer	false	false
first.visited.page.field	Name of the field containing the first visited page in the session		firstVisitedPage	false	false
utm_source.field	Name of the field containing the utm_source value in the session		utm_source	false	false
utm_medium.field	Name of the field containing the utm_medium value in the session		utm_medium	false	false
utm_campaign.field	Name of the field containing the utm_campaign value in the session		utm_campaign	false	false
utm_content.field	Name of the field containing the utm_content value in the session		utm_content	false	false
utm_term.field	Name of the field containing the utm_term value in the session		utm_term	false	false
source_of_traffic.prefix	Prefix for the source of the traffic related fields		source_of_traffic	false	false
source_of_traffic.hierarchical	Should the additional source of trafic information fields be added under a hierarchical father field or not.		false	false	false
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
cache.service	Name of the cache service to use.		null	false	false
cache.validity.timeout	Timeout validity (in seconds) of an entry in the cache.		0	false	false
debug	If true, an additional debug field is added. If the source info fields prefix is X, a debug field named X_from_cache contains a boolean value to indicate the origin of the source fields. The default value for this property is false (debug is disabled).		false	false	false
es.index	Name of the ES index containing the list of search engines and social network.		null	false	false
es.type	Name of the ES type to use.		default	false	false
es.search_engine.field	Name of the ES field used to specify that the domain is a search engine.		search_engine	false	false
es.social_network.field	Name of the ES field used to specify that the domain is a social network.		social_network	false	false

Extra informations

IpToFqdn

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToFqdn

Tags

dns, ip, fqdn, domain, address, fqhn, reverse, resolution, enrich

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1524: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
ip.address.field	The name of the field containing the ip address to use.		null	false	false
fqdn.field	The field that will contain the full qualified domain name corresponding to the ip address.		null	false	false
overwrite.fqdn.field	If the field should be overwritten when it already exists.		false	false	false
cache.service	The name of the cache service to use.		null	false	false
cache.max.time	The amount of time, in seconds, for which a cached FQDN value is valid in the cache service. After this delay, the next new request to translate the same IP into FQDN will trigger a new reverse DNS request and the result will overwrite the entry in the cache. This allows two things: if the IP was not resolved into a FQDN, this will get a chance to obtain a FQDN if the DNS system has been updated, if the IP is resolved into a FQDN, this will allow to be more accurate if the DNS system has been updated. A value of 0 seconds disables this expiration mechanism. The default value is 84600 seconds, which corresponds to new requests triggered every day if a record with the same IP passes every day in the processor.		84600	false	false
resolution.timeout	The amount of time, in milliseconds, to wait at most for the resolution to occur. This avoids to block the stream for too much time. Default value is 1000ms. If the delay expires and no resolution could occur before, the FQDN field is not created. A special value of 0 disables the logisland timeout and the resolution request may last for many seconds if the IP cannot be translated into a FQDN by the underlying operating system. In any case, whether the timeout occurs in logisland of in the operating system, the fact that a timeout occurs is kept in the cache system so that a resolution request for the same IP will not occur before the cache entry expires.		1000	false	false
debug	If true, some additional debug fields are added. If the FQDN field is named X, a debug field named X_os_resolution_time_ms contains the resolution time in ms (using the operating system, not the cache). This field is added whether the resolution occurs or time is out. A debug field named X_os_resolution_timeout contains a boolean value to indicate if the timeout occurred. Finally, a debug field named X_from_cache contains a boolean value to indicate the origin of the FQDN field. The default value for this property is false (debug is disabled.		false	false	false

Extra informations

Translates an IP address into a FQDN (Fully Qualified Domain Name). An input field from the record has the IP as value. An new field is created and its value is the FQDN matching the IP address. The resolution mechanism is based on the underlying operating system. The resolution request may take some time, specially if the IP address cannot be translated into a FQDN. For these reasons this processor relies on the logisland cache service so that once a resolution occurs or not, the result is put into the cache. That way, the real request for the same IP is not re-triggered during a certain period of time, until the cache entry expires. This timeout is configurable but by default a request for the same IP is not triggered before 24 hours to let the time to the underlying DNS system to be potentially updated.

IpToGeo

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

Module

com.hurence.logisland:logisland-processor-enrichment:1.4.0

Class

com.hurence.logisland.processor.enrichment.IpToGeo

Tags

geo, enrich, ip

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1525: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
ip.address.field	The name of the field containing the ip address to use.		null	false	false
iptogeo.service	The reference to the IP to Geo service to use.		null	false	false
geo.fields	Comma separated list of geo information fields to add to the record. Defaults to '*', which means to include all available fields. If a list of fields is specified and the data is not available, the geo field is not created. The geo fields are dependant on the underlying defined Ip to Geo service. The currently only supported type of Ip to Geo service is the Maxmind Ip to Geo service. This means that the currently supported list of geo fields is the following:**continent**: the identified continent for this IP address. **continent_code**: the identified continent code for this IP address. **city**: the identified city for this IP address. **latitude**: the identified latitude for this IP address. **longitude**: the identified longitude for this IP address. **location**: the identified location for this IP address, defined as Geo-point expressed as a string with the format: 'latitude,longitude'. **accuracy_radius**: the approximate accuracy radius, in kilometers, around the latitude and longitude for the location. **time_zone**: the identified time zone for this IP address. **subdivision_N**: the identified subdivision for this IP address. N is a one-up number at the end of the attribute name, starting with 0. **subdivision_isocode_N**: the iso code matching the identified subdivision_N. **country**: the identified country for this IP address. **country_isocode**: the iso code for the identified country for this IP address. **postalcode**: the identified postal code for this IP address. **lookup_micros**: the number of microseconds that the geo lookup took. The Ip to Geo service must have the lookup_micros property enabled in order to have this field available.		•	false	false
geo.hierarchical	Should the additional geo information fields be added under a hierarchical father field or not.		true	false	false
geo.hierarchical.suffix	Suffix to use for the field holding geo information. If geo.hierarchical is true, then use this suffix appended to the IP field name to define the father field name. This may be used for instance to distinguish between geo fields with various locales using many Ip to Geo service instances.		_geo	false	false
geo.flat.suffix	Suffix to use for geo information fields when they are flat. If geo.hierarchical is false, then use this suffix appended to the IP field name but before the geo field name. This may be		_geo_	false	false

Extra informations

Looks up geolocation information for an IP address. The attribute that contains the IP address to lookup must be provided in the **ip.address.field** property. By default, the geo information are put in a hierarchical structure. That is, if the name of the IP field is 'X', then the the geo attributes added by enrichment are added under a father field named X_geo. "_geo" is the default hierarchical suffix that may be changed with the **geo.hierarchical.suffix** property. If one wants to put the geo fields at the same level as the IP field, then the **geo.hierarchical** property should be set to false and then the geo attributes are created at the same level as him with the naming pattern X_geo_<geo_field>. "_geo_" is the default flat suffix but this may be changed with the **geo.flat.suffix** property. The IpToGeo processor requires a reference to an Ip to Geo service. This must be defined in the **iptogeo.service** property. The added geo fields are dependant on the underlying Ip to Geo service. The **geo.fields** property must contain the list of geo fields that should be created if data is available for the IP to resolve. This property defaults to "*" which means to add every available fields. If one only wants a subset of the fields, one must define a comma separated list of fields as a value for the **geo.fields** property. The list of the available geo fields is in the description of the **geo.fields** property.

ParseNetworkPacket

The ParseNetworkPacket processor is the LogIsland entry point to parse network packets captured either off-the-wire (stream mode) or in pcap format (batch mode). In batch mode, the processor decodes the bytes of the incoming pcap record, where a Global header followed by a sequence of [packet header, packet data] pairs are stored. Then, each incoming pcap event is parsed into n packet records. The fields of packet headers are then extracted and made available in dedicated record fields. See the Capturing Network packets tutorial for an example of usage of this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.networkpacket.ParseNetworkPacket

Tags

PCap, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1526: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug.		false	false	false
flow.mode	Flow Mode. Indicate whether packets are provided in batch mode (via pcap files) or in stream mode (without headers). Allowed values are batch and stream.	batch, stream	null	false	false

Extra informations

No additional information is provided

BulkAddElasticsearch

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1527: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
default.index	The name of the index to insert into		null	false	**true**
default.type	The type of this document (used by Elasticsearch for indexing and searching)		null	false	**true**
timebased.index	do we add a date suffix	no (no date added to default index), today (today's date added to default index), yesterday (yesterday's date added to default index)	no	false	false
es.index.field	the name of the event field containing es index name => will override index value if set		null	false	false
es.type.field	the name of the event field containing es doc type => will override type value if set		null	false	false

Extra informations

Indexes the content of a Record in Elasticsearch using elasticsearch's bulk processor.

MultiGetElasticsearch

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.

- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.MultiGetElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1528: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
es.index.field	the name of the incoming records field containing es index name to use in multiget query.		null	false	false
es.type.field	the name of the incoming records field containing es type name to use in multiget query		null	false	false
es.ids.field	the name of the incoming records field containing es document Ids to use in multiget query		null	false	false
es.includes.field	the name of the incoming records field containing es includes to use in multiget query		null	false	false
es.excludes.field	the name of the incoming records field containing es excludes to use in multiget query		null	false	false

Extra informations

Retrieves a content indexed in elasticsearch using elasticsearch multiget queries. Each incoming record contains information regarding the elasticsearch multiget query that will be performed. This information is stored in record fields whose names are configured in the plugin properties (see below) :

- index (String) : name of the elasticsearch index on which the multiget query will be performed. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.

- type (String) : name of the elasticsearch type on which the multiget query will be performed. This field is not mandatory.
- ids (String) : comma separated list of document ids to fetch. This field is mandatory and should not be empty, otherwise an error output record is sent for this specific incoming record.
- includes (String) : comma separated list of patterns to filter in (include) fields to retrieve. Supports wildcards. This field is not mandatory.
- excludes (String) : comma separated list of patterns to filter out (exclude) fields to retrieve. Supports wildcards. This field is not mandatory.

Each outcoming record holds data of one elasticsearch retrieved document. This data is stored in these fields :

- index (same field name as the incoming record) : name of the elasticsearch index.
- type (same field name as the incoming record) : name of the elasticsearch type.
- id (same field name as the incoming record) : retrieved document id.
- a list of String fields containing :
  - field name : the retrieved field name
  - field value : the retrieved field value

EvaluateXPath

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

Module

com.hurence.logisland:logisland-processor-xml:1.4.0

Class

com.hurence.logisland.processor.xml.EvaluateXPath

Tags

XML, evaluate, XPath

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1529: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
source	Indicates the attribute containing the xml data to evaluate xpath against.		null	false	false
validate_dtd	Specifies whether or not the XML content should be validated against the DTD.	true, false	true	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1530: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
An attribute	An XPath expression	the attribute is set to the result of the XPath Expression.		null	false

Extra informations

Evaluates one or more XPaths against the content of a record. The results of those XPaths are assigned to new attributes in the records, depending on configuration of the Processor. XPaths are entered by adding user-defined properties; the name of the property maps to the Attribute Name into which the result will be placed. The value of the property must be a valid XPath expression. If the expression matches nothing, no attributes is added.

ConsolidateSession

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”,

“int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.ConsolidateSession

Tags

analytics, web, session

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1531: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		null	false	false
session.timeout	session timeout in sec		1800	false	false
sessionid.field	the name of the field containing the session id => will override default value if set		sessionId	false	false
timestamp.field	the name of the field containing the timestamp => will override default value if set		h2kTimestamp	false	false
visitedpage.field	the name of the field containing the visited page => will override default value if set		location	false	false
userid.field	the name of the field containing the userId => will override default value if set		userId	false	false
fields.to.return	the list of fields to return		null	false	false
firstVisitedPage.field	the name of the field containing the first visited page => will override default value if set		firstVisitedPage	false	false
lastVisitedPage.field	the name of the field containing the last visited page => will override default value if set		lastVisitedPage	false	false
isSessionActive.field	the name of the field stating whether the session is active or not => will override default value if set		is_sessionActive	false	false
sessionDuration.field	the name of the field containing the session duration => will override default value if set		sessionDuration	false	false
eventsCounter.field	the name of the field containing the session duration => will override default value if set		eventsCounter	false	false
firstEventDateTime.field	the name of the field containing the date of the first event => will override default value if set		firstEventDateTime	false	false
lastEventDateTime.field	the name of the field containing the date of the last event => will override default value if set		lastEventDateTime	false	false
sessionInactivityDuration.field	the name of the field containing the session inactivity duration => will override default value if set		sessionInactivityDuration	false	false

Extra informations

The ConsolidateSession processor is the Logisland entry point to get and process events from the Web Analytics.As an example here is an incoming event from the Web Analytics:

“fields”: [{ “name”: “timestamp”, “type”: “long” },{ “name”: “remoteHost”, “type”: “string”},{ “name”: “record_type”, “type”: [“null”, “string”], “default”: null },{ “name”: “record_id”, “type”: [“null”, “string”], “default”: null },{ “name”: “location”, “type”: [“null”, “string”], “default”: null },{ “name”: “hitType”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventCategory”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventAction”, “type”: [“null”, “string”], “default”: null },{ “name”: “eventLabel”, “type”: [“null”, “string”], “default”: null },{ “name”: “localPath”, “type”: [“null”, “string”], “default”: null },{ “name”: “q”, “type”: [“null”, “string”], “default”: null },{ “name”: “n”, “type”: [“null”, “int”], “default”: null },{ “name”: “referer”, “type”: [“null”, “string”], “default”: null },{ “name”: “viewportPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”:

“viewportPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelWidth”, “type”: [“null”, “int”], “default”: null },{ “name”: “screenPixelHeight”, “type”: [“null”, “int”], “default”: null },{ “name”: “partyId”, “type”: [“null”, “string”], “default”: null },{ “name”: “sessionId”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageViewId”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_newSession”, “type”: [“null”, “boolean”],”default”: null },{ “name”: “userAgentString”, “type”: [“null”, “string”], “default”: null },{ “name”: “pageType”, “type”: [“null”, “string”], “default”: null },{ “name”: “UserId”, “type”: [“null”, “string”], “default”: null },{ “name”: “B2Bunit”, “type”: [“null”, “string”], “default”: null },{ “name”: “pointOfService”, “type”: [“null”, “string”], “default”: null },{ “name”: “companyID”, “type”: [“null”, “string”], “default”: null },{ “name”: “GroupCode”, “type”: [“null”, “string”], “default”: null },{ “name”: “userRoles”, “type”: [“null”, “string”], “default”: null },{ “name”: “is_PunchOut”, “type”: [“null”, “string”], “default”: null }]The ConsolidateSession processor groups the records by sessions and compute the duration between now and the last received event. If the distance from the last event is beyond a given threshold (by default 30mn), then the session is considered closed.The ConsolidateSession is building an aggregated session object for each active session.This aggregated object includes: - The actual session duration. - A boolean representing wether the session is considered active or closed. Note: it is possible to ressurect a session if for instance an event arrives after a session has been marked closed. - User related infos: userId, B2Bunit code, groupCode, userRoles, companyId - First visited page: URL - Last visited page: URL The properties to configure the processor are: - sessionid.field: Property name containing the session identifier (default: sessionId). - timestamp.field: Property name containing the timestamp of the event (default: timestamp). - session.timeout: Timeframe of inactivity (in seconds) after which a session is considered closed (default: 30mn). - visitedpage.field: Property name containing the page visited by the customer (default: location). - fields.to.return: List of fields to return in the aggregated object. (default: N/A)

See Also:

‘com.hurence.logisland.processor.webanalytics.IncrementalWebSession‘_

DetectOutliers

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)
- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

Module

com.hurence.logisland:logisland-processor-outlier-detection:1.4.0

Class

com.hurence.logisland.processor.DetectOutliers

Tags

analytic, outlier, record, iot, timeseries

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1532: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
value.field	the numeric field to get the value		record_value	false	false
time.field	the numeric field to get the value		record_time	false	false
output.record.type	the output type of the record		alert_match	false	false
rotation.policy.type		by_amount, by_time, never	by_amount	false	false
rotation.policy.amount			100	false	false
rotation.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
chunking.policy.type		by_amount, by_time, never	by_amount	false	false
chunking.policy.amount			100	false	false
chunking.policy.unit		milliseconds, seconds, hours, days, months, years, points	points	false	false
sketchy.outlier.algorithm		SKETCHY_MOVING_MAD	SKETCHY_MOVING_MAD	false	false
batch.outlier.algorithm		RAD	RAD	false	false
global.statistics.min	minimum value		null	false	false
global.statistics.max	maximum value		null	false	false
global.statistics.mean	mean value		null	false	false
global.statistics.stddev	standard deviation value		null	false	false
zscore.cutoffs.normal	zscore Cutoffs level for normal outlier		0.000000000000001	false	false
zscore.cutoffs.moderate	zscore Cutoffs level for moderate outlier		1.5	false	false
zscore.cutoffs.severe	zscore Cutoffs level for severe outlier		10.0	false	false
zscore.cutoffs.notEnoughData	zscore Cutoffs level for notEnoughData outlier		100	false	false
smooth	do smoothing ?		false	false	false
decay	the decay		0.1	false	false
min.amount.to.predict	minAmountToPredict		100	false	false
min_zscore_percentile	minZscorePercentile		50.0	false	false
reservoir_size	the size of points reservoir		100	false	false
rpca.force.diff	No Description Provided.		null	false	false
rpca.lpenalty	No Description Provided.		null	false	false
rpca.min.records	No Description Provided.		null	false	false
rpca.spenalty	No Description Provided.		null	false	false
rpca.threshold	No Description Provided.		null	false	false

Extra informations

Outlier Analysis: A Hybrid Approach

In order to function at scale, a two-phase approach is taken

For every data point

- Detect outlier candidates using a robust estimator of variability (e.g. median absolute deviation) that uses distributional sketching (e.g. Q-trees)

- Gather a biased sample (biased by recency)
- Extremely deterministic in space and cheap in computation

For every outlier candidate

- Use traditional, more computationally complex approaches to outlier analysis (e.g. Robust PCA) on the biased sample
- Expensive computationally, but run infrequently

This becomes a data filter which can be attached to a timeseries data stream within a distributed computational framework (i.e. Storm, Spark, Flink, NiFi) to detect outliers.

EnrichRecordsElasticsearch

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

Module

com.hurence.logisland:logisland-processor-elasticsearch:1.4.0

Class

com.hurence.logisland.processor.elasticsearch.EnrichRecordsElasticsearch

Tags

elasticsearch

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1533: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
elasticsearch.client.service	The instance of the Controller Service to use for accessing Elasticsearch.		null	false	false
record.key	The name of field in the input record containing the document id to use in ES multiget query		null	false	**true**
es.index	The name of the ES index to use in multiget query.		null	false	**true**
es.type	The name of the ES type to use in multiget query.		_doc	false	**true**
es.includes.field	The name of the ES fields to include in the record.		•	false	**true**
es.excludes.field	The name of the ES fields to exclude.		N/A	false	false
cache.service	The instance of the Cache Service to use (optional).		null	false	false

Extra informations

Enrich input records with content indexed in elasticsearch using multiget queries. Each incoming record must be possibly enriched with information stored in elasticsearch. Each outcoming record holds at least the input record plus potentially one or more fields coming from of one elasticsearch document.

ExcelExtract

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

Module

com.hurence.logisland:logisland-processor-excel:1.4.0

Class

com.hurence.logisland.processor.excel.ExcelExtract

Tags

excel, processor, poi

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1534: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
sheets	Comma separated list of Excel document sheet names that should be extracted from the excel document. If this property is left blank then all of the sheets will be extracted from the Excel document. You can specify regular expressions. Any sheets not specified in this value will be ignored.			false	false
skip.columns	Comma delimited list of column numbers to skip. Use the columns number and not the letter designation. Use this to skip over columns anywhere in your worksheet that you don't want extracted as part of the record.			false	false
field.names	The comma separated list representing the names of columns of extracted cells. Order matters! You should use either field.names either field.row.header but not both together.		null	false	false
skip.rows	The row number of the first row to start processing.Use this to skip over rows of data at the top of your worksheet that are not part of the dataset.Empty rows of data anywhere in the spreadsheet will always be skipped, no matter what this value is set to.		0	false	false
record.type	Default type of record		excel_record	false	false
field.row.header	If set, field names mapping will be extracted from the specified row number. You should use either field.names either field.row.header but not both together.		null	false	false

Extra informations

Consumes a Microsoft Excel document and converts each worksheet's line to a structured record. The processor is assuming to receive raw excel file as input record.

MatchIP

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchIP

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1535: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1536: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

IP address Query matching (using **'Luwak <http://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/>)'_**

You can use this processor to handle custom events matching IP address (CIDR) The record sent from a matching an IP address record is tagged appropriately.

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

---

MatchQuery

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

> **Warning:** don't forget to set numeric fields property to handle correctly numeric ranges queries

Module

com.hurence.logisland:logisland-processor-querymatcher:1.4.0

Class

com.hurence.logisland.processor.MatchQuery

Tags

analytic, percolator, record, record, query, lucene

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1537: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
numeric.fields	a comma separated string of numeric field to be matched		null	false	false
output.record.type	the output type of the record		alert_match	false	false
record.type.update.policy	Record type update policy		overwrite	false	false
policy.onmatch	the policy applied to match events: 'first' (default value) match events are tagged with the name and value of the first query that matched;'all' match events are tagged with all names and values of the queries that matched.		first	false	false
policy.onmiss	the policy applied to miss events: 'discard' (default value) drop events that did not match any query;'forward' include also events that did not match any query.		discard	false	false

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1538: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
query	some Lucene query	generate a new record when this query is matched		null	**true**

Extra informations

Query matching based on Luwak

you can use this processor to handle custom events defined by lucene queries a new record is added to output each time a registered query is matched

A query is expressed as a lucene query against a field like for example:

```
message:'bad exception'
error_count:[10 TO *]
bytes_out:5000
user_name:tom*
```

Please read the Lucene syntax guide for supported operations

Warning: don't forget to set numeric fields property to handle correctly numeric ranges queries.

ParseBroEvent

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{

“conn”: {

“id.resp_p”: 9092,

“resp_pkts”: 0,

“resp_ip_bytes”: 0,

“local_orig”: true,

“orig_ip_bytes”: 0,

“orig_pkts”: 0,

“missed_bytes”: 0,

“history”: “Cc”,

“tunnel_parents”: [],

“id.orig_p”: 56762,

“local_resp”: true,

```
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.bro.ParseBroEvent

Tags

bro, security, IDS, NIDS

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1539: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false

Extra informations

The ParseBroEvent processor is the Logisland entry point to get and process Bro events. The Bro-Kafka plugin should be used and configured in order to have Bro events sent to Kafka. See the Bro/Logisland tutorial for an example of usage for this processor. The ParseBroEvent processor does some minor pre-processing on incoming Bro events from the Bro-Kafka plugin to adapt them to Logisland.

Basically the events coming from the Bro-Kafka plugin are JSON documents with a first level field indicating the type of the event. The ParseBroEvent processor takes the incoming JSON document, sets the event type in a record_type field and sets the original sub-fields of the JSON event as first level fields in the record. Also any dot in a field name is transformed into an underscore. Thus, for instance, the field id.orig_h becomes id_orig_h. The next processors in the stream can then process the Bro events generated by this ParseBroEvent processor.

As an example here is an incoming event from Bro:

{
“conn”: {
“id.resp_p”: 9092,
“resp_pkts”: 0,
“resp_ip_bytes”: 0,
“local_orig”: true,
“orig_ip_bytes”: 0,
“orig_pkts”: 0,
“missed_bytes”: 0,
“history”: “Cc”,
“tunnel_parents”: [],

```
            "id.orig_p": 56762,
            "local_resp": true,
            "uid": "Ct3Ms01I3Yc6pmMZx7",
            "conn_state": "OTH",
            "id.orig_h": "172.17.0.2",
            "proto": "tcp",
            "id.resp_h": "172.17.0.3",
            "ts": 1487596886.953917
        }
    }
```


It gets processed and transformed into the following Logisland record by the ParseBroEvent processor:

"@timestamp": "2017-02-20T13:36:32Z"

"record_id": "6361f80a-c5c9-4a16-9045-4bb51736333d"

"record_time": 1487597792782

"record_type": "conn"

"id_resp_p": 9092

"resp_pkts": 0

"resp_ip_bytes": 0

"local_orig": true

"orig_ip_bytes": 0

"orig_pkts": 0

"missed_bytes": 0

"history": "Cc"

"tunnel_parents": []

"id_orig_p": 56762

"local_resp": true

"uid": "Ct3Ms01I3Yc6pmMZx7"

"conn_state": "OTH"

"id_orig_h": "172.17.0.2"

"proto": "tcp"

"id_resp_h": "172.17.0.3"

"ts": 1487596886.953917

ParseNetflowEvent

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

Module

com.hurence.logisland:logisland-processor-cyber-security:1.4.0

Class

com.hurence.logisland.processor.netflow.ParseNetflowEvent

Tags

netflow, security

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1540: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
debug	Enable debug. If enabled, the original JSON string is embedded in the record_value field of the record.		false	false	false
output.record.type	the output type of the record		netflowevent	false	false
enrich.record	Enrich data. If enabledthe netflow record is enriched with inferred data		false	false	false

Extra informations

The Netflow V5 processor is the Logisland entry point to process Netflow (V5) events. NetFlow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic.We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards one or more flow collectors

- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are then available for analysis purpose (intrusion detection, traffic analysis...) Netflow are sent to kafka in order to be processed by logisland. In the tutorial we will simulate Netflow traffic using nfgen. this traffic will be sent to port 2055. The we rely on nifi to listen of that port for incoming netflow (V5) traffic and send them to a kafka topic. The Netflow processor could thus treat these events and generate corresponding logisland records. The following processors in the stream can then process the Netflow records generated by this processor.

RunPython

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

Module

com.hurence.logisland:logisland-processor-scripting:1.4.0

Class

com.hurence.logisland.processor.scripting.python.RunPython

Tags

scripting, python

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1541: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
script.code.imports	For inline mode only. This is the python code that should hold the import statements if required.		null	false	false
script.code.init	The python code to be called when the processor is initialized. This is the python equivalent of the init method code for a java processor. This is not mandatory but can only be used if **script.code.process** is defined (inline mode).		null	false	false
script.code.process	The python code to be called to process the records. This is the pyhton equivalent of the process method code for a java processor. For inline mode, this is the only minimum required configuration property. Using this property, you may also optionally define the **script.code.init** and **script.code.imports** properties.		null	false	false
script.path	The path to the user's python processor script. Use this property for file mode. Your python code must be in a python file with the following constraints: let's say your pyhton script is named MyProcessor.py. Then MyProcessor.py is a module file that must contain a class named MyProcessor which must inherits from the Logisland delivered class named AbstractProcessor. You can then define your code in the process method and in the other traditional methods (init...) as you would do in java in a class inheriting from the AbstractProcessor java class.		null	false	false
dependencies.path	The path to the additional dependencies for the user's python code, whether using inline or file mode. This is optional as your code may not have additional dependencies. If you defined **script.path** (so using file mode) and if **dependencies.path** is not defined, Logisland will scan a potential directory named **dependencies** in the same directory where the script file resides and if it exists, any python code located there will be loaded as dependency as needed.		null	false	false
logisland.dependencies.path	The path to the directory containing the python dependencies shipped with logisland. You should not have to tune this parameter.		null	false	false

Extra informations

!!!! WARNING !!!!

The RunPython processor is currently an experimental feature : it is delivered as is, with the current set of features and is subject to modifications in API or anything else in further logisland releases without warnings. There is no tutorial yet. If you want to play with this processor, use the python-processing.yml example and send the apache logs of the index apache logs tutorial. The debug stream processor at the end of the stream should output events in stderr file of the executors from the spark console.

This processor allows to implement and run a processor written in python. This can be done in 2 ways. Either directly defining the process method code in the **script.code.process** configuration property or poiting to an external python module script file in the **script.path** configuration property. Directly defining methods is called the inline mode whereas using a script file is called the file mode. Both ways are mutually exclusive. Whether using the inline of file mode, your python code may depend on some python dependencies. If the set of python dependencies already delivered with the Logisland framework is not sufficient, you can use the **dependencies.path** configuration property to give their location. Currently only the nltk python library is delivered with Logisland.

URIDecoder

Decode one or more field containing an URI with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URIDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1542: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLCleaner

Remove some or all query parameters from one or more field containing an uri which should be preferably encoded. If the uri is not encoded the behaviour is not defined in case the decoded uri contains '#', '?', '=', '&' which were encoded. Indeed this processor assumes that the start of query part of the uri start at the first '?' then end at the first '#' or at the end of the uri as specified by rfc3986 available at https://tools.ietf.org/html/rfc3986#section-3.4. We assume as well that key value pairs are separed by '=', and are separed by '&': exemple 'param1=value1&param2=value2'. The processor can remove also parameters that have only a name and no value. The character used to separate the key and the value '=' is configurable. The character used to separate two parameters '&' is also configurable.

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLCleaner

Tags

record, fields, url, params, param, remove, keep, query, uri, parameter, clean, decoded, raw

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1543: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
url.fields	List of fields (URL) to decode and optionnaly the output field for the url modified. Syntax should be <name>,<name:newName>,...,<name>. So fields name can not contain ',' nor ':'		null	false	false
conflict.resolution.policy	What to do when a field with the same name already exists ?	overwrite_existing (if field already exist), keep_only_old_field (keep only old field)	keep_only_old_field	false	false
url.keep.params	List of param names to keep in the input url (others will be removed). Can not be given at the same time as url.remove.params or url.remove.all		null	false	false
url.remove.params	List of param names to remove from the input url (others will be kept). Can not be given at the same time as url.keep.params or url.remove.all		null	false	false
url.remove.all	Remove all params if true.		null	false	false
parameter.separator	the character to use to separate the parameters in the query part of the uris		&	false	false
key.value.separator	the character to use to separate the parameter name from the parameter value in the query part of the uris		=	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

URLDecoder

Decode one or more field containing an URL with possibly special chars encoded ...

Module

com.hurence.logisland:logisland-processor-web-analytics:1.4.0

Class

com.hurence.logisland.processor.webanalytics.URLDecoder

Tags

record, fields, Decode

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1544: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
decode.fields	List of fields (URL) to decode		null	false	false
charset	Charset to use to decode the URL		UTF-8	false	false

Extra informations

Decode one or more field containing an URL with possibly special chars encoded.

Services

Find below the list.

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1545: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1546: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_2_4_0_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 2.4.0.

Module

com.hurence.logisland:logisland-service-elasticsearch_2_4_0-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_2_4_0_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1547: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
cluster.name	Name of the ES cluster (for example, elasticsearch_brew). Defaults to 'elasticsearch'		elasticsearch	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the		null	false	false

Extra informations

No additional information is provided

Elasticsearch_5_4_0_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 5.4.0.

Module

com.hurence.logisland:logisland-service-elasticsearch_5_4_0-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_5_4_0_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1548: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
cluster.name	Name of the ES cluster (for example, elasticsearch_brew). Defaults to 'elasticsearch'		elasticsearch	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the		null	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1549: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1550: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1551: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1552: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1553: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1554: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1555: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1556: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1557: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1558: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Solr_6_4_2_ChronixClientService

Implementation of ChronixClientService for Solr 6 4 2

Module

com.hurence.logisland:logisland-service-solr_chronix_6_4_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_4_2_ChronixClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1559: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
flush.interval	flush interval in ms		500	false	false
group.by	The field the chunk should be grouped by			false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1560: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Solr_8_ChronixClientService

Implementation of ChronixClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_chronix_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_8_ChronixClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1561: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
flush.interval	flush interval in ms		500	false	false
group.by	The field the chunk should be grouped by			false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1562: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1563: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1564: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The username to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1565: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1566: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1567: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1568: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1569: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1570: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1571: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1572: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1573: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1574: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1575: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1576: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1577: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1578: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1579: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1580: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1581: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1582: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1583: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1584: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1585: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1586: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1587: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1588: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1589: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1590: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1591: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1592: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1593: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1594: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1595: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1596: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1597: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1598: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1599: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1600: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1601: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1602: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1603: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.username	The username to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1604: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1605: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1606: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1607: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]... [;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]... [;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>... [;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1608: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1609: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1610: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1611: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1612: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1613: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1614: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1615: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1616: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1617: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1618: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1619: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1620: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1621: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1622: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1623: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1624: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1625: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1626: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1627: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1628: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1629: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1630: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1631: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1632: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1633: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1634: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1635: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1636: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1637: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1638: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1639: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1640: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1641: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1642: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1643: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1644: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1645: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1646: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]... [;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]... [;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>... [;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1647: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1648: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1649: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1650: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1651: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1652: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1653: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1654: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1655: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1656: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1657: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1658: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1659: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1660: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1661: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1662: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1663: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1664: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1665: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1666: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1667: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1668: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1669: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1670: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1671: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1672: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1673: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1674: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1675: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1676: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1677: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1678: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1679: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1680: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1681: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1682: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1683: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1684: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1685: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1686: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1687: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1688: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1689: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1690: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1691: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1692: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1693: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1694: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1695: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1696: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1697: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1698: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1699: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1700: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1701: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1702: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1703: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1704: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1705: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1706: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1707: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1708: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1709: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1710: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1711: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1712: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1713: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1714: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1715: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1716: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1717: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1718: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1719: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1720: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1721: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1722: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1723: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1724: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1725: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1726: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1727: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1728: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1729: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1730: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1731: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1732: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1733: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1734: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1735: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1736: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1737: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1738: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1739: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1740: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1741: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1742: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1743: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1744: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1745: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1746: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1747: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1748: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1749: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1750: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1751: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1752: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1753: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1754: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1755: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1756: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1757: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1758: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1759: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1760: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1761: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1762: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1763: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1764: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1765: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1766: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1767: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1768: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1769: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1770: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1771: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1772: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1773: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1774: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1775: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1776: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1777: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1778: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1779: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1780: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1781: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1782: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1783: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1784: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1785: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1786: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1787: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1788: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1789: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1790: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1791: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1792: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1793: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1794: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1795: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1796: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1797: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1798: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1799: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1800: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1801: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1802: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1803: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1804: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1805: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1806: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1807: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1808: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1809: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1810: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1811: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1812: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1813: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1814: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1815: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1816: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1817: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1818: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1819: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1820: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1821: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1822: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1823: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1824: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1825: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1826: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1827: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1828: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1829: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1830: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1831: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1832: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1833: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1834: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.3.0

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1835: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.3.0

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1836: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.3.0

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1837: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.3.0

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1838: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.3.0

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1839: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1840: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.3.0

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1841: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.3.0

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1842: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.3.0

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1843: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.3.0

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1844: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.3.0

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1845: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.3.0

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1846: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.3.0

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1847: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1848: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1849: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1850: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1851: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1852: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1853: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1854: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1855: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1856: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${’{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1857: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1858: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1859: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1860: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1861: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1862: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1863: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1864: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1865: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1866: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1867: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1868: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1869: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1870: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1871: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1872: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1873: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1874: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1875: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1876: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1877: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1878: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1879: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1880: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1881: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1882: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1883: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1884: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1885: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1886: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1887: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1888: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1889: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1890: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1891: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1892: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1893: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1894: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1895: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1896: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1897: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1898: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1899: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1900: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1901: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1902: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1903: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1904: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1905: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1906: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1907: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1908: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1909: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1910: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1911: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1912: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1913: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1914: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1915: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1916: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1917: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1918: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1919: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1920: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1921: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1922: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1923: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1924: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1925: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1926: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1927: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1928: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1929: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1930: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1931: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1932: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1933: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1934: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1935: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1936: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1937: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1938: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1939: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1940: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1941: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1942: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1943: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1944: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1945: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1946: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1947: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1948: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1949: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1950: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1951: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1952: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1953: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1954: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1955: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1956: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1957: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1958: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1959: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1960: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1961: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1962: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1963: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1964: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1965: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1966: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1967: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1968: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1969: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1970: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1971: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1972: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1973: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1974: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1975: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1976: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1977: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1978: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1979: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1980: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1981: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1982: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1983: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1984: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1985: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1986: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1987: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1988: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1989: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1990: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1991: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1992: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1993: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 1994: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1995: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 1996: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1997: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 1998: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 1999: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2000: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2001: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2002: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2003: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2004: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2005: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2006: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2007: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2008: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2009: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2010: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2011: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2012: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2013: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2014: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2015: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2016: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2017: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2018: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2019: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2020: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2021: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2022: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2023: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]... [;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]... [;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>... [;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2024: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2025: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2026: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2027: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2028: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2029: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2030: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2031: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2032: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2033: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2034: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2035: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2036: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2037: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2038: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2039: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2040: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2041: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2042: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2043: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2044: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2045: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.username	The username to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2046: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2047: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2048: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2049: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2050: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2051: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2052: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2053: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2054: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2055: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2056: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2057: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2058: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2059: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2060: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2061: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2062: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2063: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2064: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2065: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2066: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2067: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2068: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2069: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2070: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2071: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2072: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2073: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2074: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2075: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2076: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2077: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2078: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2079: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2080: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2081: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2082: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2083: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2084: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2085: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2086: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2087: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2088: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2089: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2090: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2091: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2092: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2093: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2094: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2095: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2096: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2097: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2098: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2099: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2100: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2101: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2102: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2103: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2104: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2105: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2106: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2107: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2108: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2109: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2110: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2111: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2112: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2113: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2114: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2115: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2116: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2117: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2118: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2119: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2120: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2121: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2122: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2123: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2124: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2125: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2126: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2127: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2128: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2129: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2130: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2131: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2132: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2133: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2134: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2135: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2136: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2137: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2138: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2139: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2140: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2141: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2142: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2143: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2144: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2145: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2146: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2147: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2148: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2149: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2150: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2151: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2152: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2153: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2154: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2155: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2156: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2157: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2158: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2159: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2160: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2161: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2162: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2163: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2164: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2165: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2166: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2167: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2168: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2169: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2170: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2171: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2172: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2173: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2174: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2175: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2176: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2177: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2178: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2179: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2180: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2181: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2182: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2183: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2184: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2185: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2186: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2187: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2188: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2189: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2190: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2191: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2192: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2193: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2194: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2195: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2196: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2197: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2198: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2199: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2200: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2201: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2202: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2203: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2204: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2205: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2206: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2207: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2208: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2209: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2210: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2211: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2212: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2213: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2214: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2215: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2216: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2217: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2218: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2219: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2220: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2221: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2222: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2223: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2224: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2225: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2226: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2227: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2228: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2229: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2230: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2231: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2232: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2233: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2234: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2235: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2236: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2237: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2238: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2239: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2240: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2241: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2242: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2243: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2244: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2245: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2246: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2247: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2248: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2249: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2250: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2251: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2252: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2253: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2254: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2255: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2256: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2257: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2258: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2259: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2260: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2261: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2262: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2263: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2264: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2265: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2266: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2267: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2268: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2269: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2270: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2271: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2272: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2273: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2274: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2275: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2276: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2277: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2278: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2279: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2280: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

HBase_1_1_2_ClientService

Implementation of HBaseClientService for HBase 1.1.2. This service can be configured by providing a comma-separated list of configuration files, or by specifying values for the other properties. If configuration files are provided, they will be loaded first, and the values of the additional properties will override the values from the configuration files. In addition, any user defined properties on the processor will also be passed to the HBase configuration.

Module

com.hurence.logisland:logisland-service-hbase_1_1_2-client:1.4.1

Class

com.hurence.logisland.service.hbase.HBase_1_1_2_ClientService

Tags

hbase, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2281: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
hadoop.configuration.files	A comma-separated list of Hadoop Configuration files, such as hbase-site.xml and core-site.xml for kerberos, including full paths to the files.		null	false	false
zookeeper.quorum	Comma-separated list of ZooKeeper hosts for HBase. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.client.port	The port on which ZooKeeper is accepting client connections. Required if Hadoop Configuration Files are not provided.		null	false	false
zookeeper.znode.parent	The ZooKeeper ZNode Parent value for HBase (example: /hbase). Required if Hadoop Configuration Files are not provided.		null	false	false
hbase.client.retries	The number of times the HBase client will retry connecting. Required if Hadoop Configuration Files are not provided.		3	false	false
phoenix.client.jar.location	The full path to the Phoenix client JAR. Required if Phoenix is installed on top of HBase.		null	false	**true**

Dynamic Properties

Dynamic Properties allow the user to specify both the name and value of a property.

Table 2282: dynamic-properties

Name	Value	Description	Allowable Values	Default Value	EL
The name of an HBase configuration property.	The value of the given HBase configuration property.	These properties will be set on the HBase configuration after loading any provided configuration files.		null	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2283: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2284: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2285: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2286: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2287: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2288: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2289: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2290: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2291: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2292: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2293: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2294: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2295: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2296: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2297: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2298: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2299: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2300: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2301: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2302: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2303: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2304: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2305: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2306: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2307: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2308: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2309: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2310: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2311: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2312: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2313: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2314: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

Elasticsearch_6_6_2_ClientService

Implementation of ElasticsearchClientService for Elasticsearch 6.6.2.

Module

com.hurence.logisland:logisland-service-elasticsearch_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_6_6_2_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2315: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false
ssl.context.service	The SSL Context Service used to provide client certificate information for TLS/SSL connections. This service only applies if the Shield plugin is available.		null	false	false
charset	Specifies the character set of the document		UTF-8	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2316: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2317: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2318: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2319: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Solr_6_6_2_ClientService

Implementation of ElasticsearchClientService for Solr 5.5.5.

Module

com.hurence.logisland:logisland-service-solr_6_6_2-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr_6_6_2_ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2320: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2321: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2322: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2323: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2324: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2325: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.username	The username to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2326: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Determines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2327: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2328: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2329: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2330: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2331: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2332: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2333: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2334: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2335: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2336: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2337: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2338: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2339: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2340: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2341: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2342: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2343: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2344: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2345: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2346: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2347: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2348: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2349: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2350: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2351: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2352: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The username to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2353: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]... [;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]... [;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>... [;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2354: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2355: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${‘{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2356: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2357: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2358: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2359: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2360: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2361: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.user	The user name to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2362: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2363: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2364: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id ('_id': data._id)		${'{ "_id" :"" + record_id + ""}'}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2365: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2366: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2367: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

MaxmindIpToGeoService

Implementation of the IP 2 GEO Service using maxmind lite db file

Module

com.hurence.logisland:logisland-service-ip-to-geo-maxmind:1.4.1

Class

com.hurence.logisland.service.iptogeo.maxmind.MaxmindIpToGeoService

Tags

ip, service, geo, maxmind

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2368: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
maxmind.database.uri	Path to the Maxmind Geo Enrichment Database File.		null	false	false
maxmind.database.path	Local Path to the Maxmind Geo Enrichment Database File.		null	false	false
locale	Locale to use for geo information. Defaults to 'en'.		en	false	false
lookup.time	Should the additional lookup_micros field be returned or not.		false	false	false

Extra informations

No additional information is provided

CSVKeyValueCacheService

A cache that store csv lines as records loaded from a file

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.CSVKeyValueCacheService

Tags

csv, service, cache

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2369: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
csv.format	a configuration for loading csv	default (Standard comma separated format, as for RFC4180 but allowing empty lines. Settings are: withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(true)), excel (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘,’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), excel_fr (Excel file format (using a comma as the value delimiter). Note that the actual value delimiter used by Excel is locale dependent, it might be necessary to customize this format to accom-modate to your regional settings. withDelimiter(‘;’) withQuote(“”) withRecord-Separator(“rn”) withIgnoreEmpty-Lines(false) withAllowMiss-ingColumn-Names(true)), mysql (Default	default	false	false

Extra informations

No additional information is provided

CassandraControllerService

Provides a controller service that for the moment only allows to bulkput records into cassandra.

Module

com.hurence.logisland:logisland-service-cassandra-client:1.4.1

Class

com.hurence.logisland.service.cassandra.CassandraControllerService

Tags

cassandra, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2370: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cassandra.hosts	Cassandra cluster hosts as a comma separated value list		null	false	false
cassandra.port	Cassandra cluster port		null	false	false
cassandra.with-ssl	If this property is true, use SSL. Default is no SSL (false).		false	false	false
cassandra.with-credentials	If this property is true, use credentials. Default is no credentials (false).		false	false	false
cassandra.credentials.username	The username to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
cassandra.credentials.password	The user password to use for authentication. cassandra.with-credentials must be true for that property to be used.		null	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
flush.interval	flush interval in ms		500	false	false

Extra informations

No additional information is provided

InfluxDBControllerService

Provides a controller service that for the moment only allows to bulkput records into influxdb.

Module

com.hurence.logisland:logisland-service-influxdb-client:1.4.1

Class

com.hurence.logisland.service.influxdb.InfluxDBControllerService

Tags

influxdb, service, time series

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2371: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	El
influxdb.url	InfluxDB connection url		null	false	false
influxdb.user	The user name to use for authentication.		null	false	false
influxdb.database	InfluxDB database name		null	false	false
influxdb.password	The user password to use for authentication.		null	false	false
influxdb.tags	List of tags for each supported measurement. Syntax: <measurement>:<tag>[,<tag>]...[;<measurement>:<tag>,[<tag>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 tags: core1 and core2 and the mem measurement has 1 tag: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_fields_but_explicit_tags.		null	false	false
influxdb.fields	List of fields for each supported measurement. Syntax: <measurement>:<field>[,<field>]...[;<measurement>:<field>,[<field>]]... Example: cpu:core1,core2;mem:used : in this example, the cpu measurement has 2 fields: core1 and core2 and the mem measurement has 1 field: used. This must only be set if configuration mode is explicit_tags_and_fields or all_as_tags_but_explicit_fields.		null	false	false
influxdb.configuration.mode	Defines the way fields and tags are chosen from the logisland record. Possible values and meaning: explicit_tags_and_fields: only logisland record fields listed in influxdb.tags and influxdb.fields will be inserted into InfluxDB with the explicit type. all_as_fields: all available logisland record fields will be inserted into InfluxDB as fields. all_as_tags_but_explicit_fields: all available logisland record fields will be inserted into InfluxDB as tags except those listed in influxdb.fields that will be inserted into InfluxDB as fields. all_as_fields_but_explicit_tags: all available logisland record fields will be inserted into InfluxDB as fields except those listed in influxdb.tags that will be inserted into InfluxDB as tags	explicit_tags_and_fields, all_as_fields, all_as_fields_but_explicit_tags, all_as_tags_but_explicit_fields	null	false	false
influxdb.consistency_level	Determines the consistency level used to write points into InfluxDB. Possible values are: ANY, ONE, QUORUMand ALL. Default value is ANY. This is only useful when using a clustered InfluxDB infrastructure.	ANY, ONE, QUORUM, ALL	ANY	false	false
influxdb.retention_policy	Determines the name of the retention policy to use. Defaults to autogen. The defined retention policy must already be defined in the InfluxDB server.		autogen	false	false
influxdb.timefield	Time field for each supported measurement. Syntax: <measurement>:<field>,<format>...[;<measurement>:<field>,<format>]... With format being any constant defined in java.util.concurrent.TimeUnit		null	false	false

Extra informations

No additional information is provided

LRUKeyValueCacheService

A controller service for caching data by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-inmemory-cache:1.4.1

Class

com.hurence.logisland.service.cache.LRUKeyValueCacheService

Tags

cache, service, key, value, pair, LRU

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2372: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
cache.size	The maximum number of element in the cache.		16384	false	false

Extra informations

No additional information is provided

MongoDBControllerService

Provides a controller service that wraps most of the functionality of the MongoDB driver.

Module

com.hurence.logisland:logisland-service-mongodb-client:1.4.1

Class

com.hurence.logisland.service.mongodb.MongoDBControllerService

Tags

mongo, mongodb, service

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property supports the Expression Language .

Table 2373: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
mongo.uri	MongoURI, typically of the form: mongodb://host1[:port1][,host2[:port2],...]		null	false	**true**
mongo.db.name	The name of the database to use		null	false	**true**
mongo.collection.name	The name of the collection to use		null	false	**true**
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
mongo.bulk.mode	Bulk mode (insert or upsert)	insert (Insert records whose key must be unique), upsert (Insert records if not already existing or update the record if already existing)	insert	false	false
flush.interval	flush interval in ms		500	false	false
mongo.write.concern	The write concern to use	ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_ACKNOWLEDGED, MAJORITY	ACKNOWLEDGED	false	false
mongo.bulk.upsert.condition	A custom condition for the bulk upsert (Filter for the bulkwrite). If not specified the standard condition is to match same id (‘_id’: data._id)		${’{ “_id” :’” + record_id + ‘”}’}	false	**true**

Extra informations

No additional information is provided

RedisKeyValueCacheService

A controller service for caching records by key value pair with LRU (last recently used) strategy. using LinkedHashMap

Module

com.hurence.logisland:logisland-service-redis:1.4.1

Class

com.hurence.logisland.redis.service.RedisKeyValueCacheService

Tags

cache, service, key, value, pair, redis

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2374: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
redis.mode	The type of Redis being communicated with - standalone, sentinel, or clustered.	standalone (A single standalone Redis instance.), sentinel (Redis Sentinel which provides high-availability. Described further at https://redis.io/topics/sentinel), cluster (Clustered Redis which provides sharding and replication. Described further at https://redis.io/topics/cluster-spec)	standalone	false	false
connection.string	The connection string for Redis. In a standalone instance this value will be of the form hostname:port. In a sentinel instance this value will be the comma-separated list of sentinels, such as host1:port1,host2:port2,host3:port3. In a clustered instance this value will be the comma-separated list of cluster masters, such as host1:port,host2:port,host3:port.		null	false	false
database.index	The database index to be used by connections created from this connection pool. See the databases property in redis.conf, by default databases 0-15 will be available.		0	false	false
communication.timeout	The timeout to use when attempting to communicate with Redis.		10 seconds	false	false
cluster.max.redirects	The maximum number of redirects that can be performed when clustered.		5	false	false
sentinel.master	The name of the sentinel master, require when Mode is set to Sentinel		null	false	false
password	The password used to authenticate to the Redis server. See the requirepass property in redis.conf.		null	**true**	false
pool.max.total	The maximum number of connections that can be allocated by the pool (checked out to clients, or idle awaiting checkout). A negative value indicates that there is no limit.		8	false	false
pool.max.idle	The maximum number of idle connections that can be held in the pool, or a negative value if there is no limit.		8	false	false
pool.min.idle	The target for the minimum number of idle connections to maintain in the pool. If the configured value of Min Idle is greater than the configured value for Max Idle, then the value of Max Idle will be used instead.		0	false	false
pool.block.when.exhausted	Whether or not clients should block and wait when trying to obtain a connection from the pool when the pool has no available connections. Setting this to false means an error will occur immediately when a client requests a connection and none are avail-	true, false	true	false	false

Extra informations

No additional information is provided

Elasticsearch_7_x_ClientService

Implementation of ElasticsearchClientService for ElasticSearch 7.x. Note that although Elasticsearch 7.x still accepts type information, this implementation will ignore any type usage and will only work at the index level to be already compliant with the ElasticSearch 8.x version that will completely remove type usage.

Module

com.hurence.logisland:logisland-service-elasticsearch_7_x-client:1.4.1

Class

com.hurence.logisland.service.elasticsearch.Elasticsearch_7_x_ClientService

Tags

elasticsearch, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values, and whether a property is considered "sensitive"..

Table 2375: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
backoff.policy	strategy for retrying to execute requests in bulkRequest	noBackoff (when a request fail there won't be any retry.), constantBackoff (wait a fixed amount of time between retries, using user put retry number and throttling delay), exponentialBackoff (time waited between retries grow exponentially, using user put retry number and throttling delay), defaultExponentialBackoff (time waited between retries grow exponentially, using es default parameters)	defaultExponentialBackoff	false	false
throttling.delay	number of time we should wait between each retry (in milliseconds)		500	false	false
num.retry	number of time we should try to inject a bulk into es		3	false	false
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
flush.interval	flush interval in sec		5	false	false
concurrent.requests	setConcurrentRequests		2	false	false
ping.timeout	The ping timeout used to determine when a node is unreachable. For example, 5s (5 seconds). If non-local recommended is 30s		5s	false	false
sampler.interval	How often to sample / ping the nodes listed and connected. For example, 5s (5 seconds). If non-local recommended is 30s.		5s	false	false
username	Username to access the Elasticsearch cluster		null	false	false
password	Password to access the Elasticsearch cluster		null	**true**	false
enable.ssl	Whether to enable (true) TLS/SSL connections or not (false). This can for instance be used with opendistro. Defaults to false. Note that the current implementation does try to validate the server certificate.		false	false	false
shield.location	Specifies the path to the JAR for the Elasticsearch Shield plugin. If the Elasticsearch cluster has been secured with the Shield plugin, then the Shield plugin JAR must also be available to this processor. Note: Do NOT place the Shield JAR into NiFi's lib/ directory, doing so will prevent the Shield plugin from being loaded.		null	false	false
hosts	ElasticSearch Hosts, which should be comma separated and colon for hostname/port host1:port,host2:port,.... For example testcluster:9300.		null	false	false

Extra informations

No additional information is provided

Solr8ClientService

Implementation of SolrClientService for Solr 8

Module

com.hurence.logisland:logisland-service-solr_8-client:1.4.1

Class

com.hurence.logisland.service.solr.Solr8ClientService

Tags

solr, client

Properties

In the list below, the names of required properties appear in **bold**. Any other properties (not in bold) are considered optional. The table also indicates any default values.

Table 2376: allowable-values

Name	Description	Allowable Values	Default Value	Sensitive	EL
batch.size	The preferred number of Records to setField to the database in a single transaction		1000	false	false
bulk.size	bulk size in MB		5	false	false
solr.cloud	is slor cloud enabled		false	false	false
solr.collection	name of the collection to use		null	false	false
solr.connection.string	zookeeper quorum host1:2181,host2:2181 for solr cloud or http address of a solr core		localhost:8983/solr	false	false
solr.concurrent.requests	setConcurrentRequests		2	false	false
flush.interval	flush interval in ms		500	false	false
schema.update_timeout	Schema update timeout interval in s		15	false	false

Extra informations

No additional information is provided

1.1.2 Dynamic properties

Overview

You use components to run jobs in logisland that manipulate records. Those components use properties that you specify in the job configuration file. Some of them are defined in advance by the component's developer. They got a name and you have to use it to define these properties. We call those properties *static properties*.

Some components support dynamic *properties*. When this is the case, any properties specified in job conf for this component that is not a static property will be used as a dynamic property instead of throwing an error for a bad configuration.

In this section we will talk about those properties and how you can use them.

Structure of a dynamic properties

Dynamic properties are really just like static properties but build on the fly. It allow to use both the name and the value of the property by the developer. For example instead of specifying :

```
record.name: myName
record.value: myValue
```

You could specify :

```
myName: myValue
```

The advantage is that you can have any number of dynamic property whereas you have to specify in advance all static properties...

Usage of a dynamic properties

You can check the documentation of com.hurence.logisland.processor.AddFields processor that we will use in those example.

Adding a field which is concatenation of two others using '_' as joining string

set those dynamic properties in AddFields processor :

- concat2fields : value1
- my_countries : 3
- my_countries.type : INT

Then records processed by this processor would have 2 more fields out of this processors:

- field 'concat2fields' of type String with value 'value1'
- field 'my_countries' of type Int with value '3'

By default if no type is specified by a dynamic property it use a type of String or the same type as old value if field already existed and you choose an overwrite policy.

See com.hurence.logisland.processor.AddFields processor doc fore more information.

Conclusion

As you can see dynamic properties are very flexible but it's usage is very dependent of the implementation of the component's developer.

1.1.3 Expression Language

Overview

All data in Logisland is represented by an abstraction called a Record. Those records contains fields of different types.

You use components to run jobs in logisland that manipulate those records. Those components use properties that you specify in the job configuration file. Some of them support the expression language (EL). In this section we will talk about those properties and how you can use them.

Structure of a Logisland Expression

The Logisland expression Language always begins with the start delimiter *${* and ends with the end delimiter *}*. Between the start and end delimiters is the text of the expression itself. In its most basic form, the expression can consist of just a record field name. For example, *${name}* will return the value of the field *name* of the record used.

The use of the property depends on the implementation of the components ! Indeed it is the component that decide to evaluate your Logisland expression with which Record.

For example the AddField processor use Logisland expression in its dynamic properties.

- The key representing the name of the field to add.
- The value can be a Logisland expression that will be used to calculate the value of the new field. In this expression you can use fields value of the current Record because it is passed as context of the Logisland expression by this processor.

So be sure to carefully read description of the properties to understand how it will be evaluated and for what purpose.

We are currently using the **mvel** language which you can check documentation here.

Note: If you want to be able to use another ScriptEngine than mvel (javascript for example). You can open an issue to ask this feature. Feel free to make a Pull request as well to implement this new feature.

We have implemented some example as unit test as well if you want to check in the code source, the class is **com.hurence.logisland.component.TestInterpretedPropertyValueWithMvelEngine** in the module **com.hurence.logisland:logisland-api**.

Otherwise we will show you some simple examples using the AddField processor in next Section.

Usage of a Logisland Expression

You can check the documentation of com.hurence.logisland.processor.AddFields processor that we will use in those example.

Adding a field which is concatenation of two others using '_' as joining string

set those dynamic properties in AddFields processor :

- concat2fields : ${field1 + "_" + field2}
- my_countries : ${["france", "allemagne"]}
- my_countries.type : array
- my_employees_by_countries : ${["france" : 100, "allemagne" : 50]}
- my_employees_by_countries.type : map

Then if in input of this processor there is records with fields : field1=value1 and field2=value2, it would have 3 more fields once out of this processor:

- field 'concat2fields' of type String with value 'value1_value2'
- field 'my_countries' of type Array containing values 'france' and 'allemagne'
- field 'my_employees_by_countries' of type Map with key value pairs "france" : 100 and "allemagne" : 50

By default if no type is specified by a dynamic property it use a type of String or the same type as old value if field already existed and you choose an overwrite policy.

See com.hurence.logisland.processor.AddFields processor doc for more information.

Conclusion

As you can see the language expression is very flexible but it's usage is very dependent of the implementation of the component's developer.

1.2 Tutorials

Chat with us on Gitter

Download the latest release build and unzip on an edge node.

Contents:

1.2.1 Prerequisites

There are two main ways to launch a logisland job :

- within Docker containers
- within an Hadoop distribution (Cloudera, Hortonworks, ...)

1. Trough a Docker container (testing way)

Logisland is packaged as a Docker container that you can build yourself or pull from Docker Hub.

To facilitate integration testing and to easily run tutorials, you can use *docker-compose* with the followings :

- docker-compose.yml.

Once you have these file you can run a *docker-compose* command to launch all the needed services (zookeeper, kafka, es, kibana, redis and logisland). (You can remove the services that you do not need depending on tutorial).

Elasticsearch on docker needs a special tweak as described here

```
# set vm.max_map_count kernel setting for elasticsearch
sudo sysctl -w vm.max_map_count=262144

#
cd /tmp
wget https://raw.githubusercontent.com/Hurence/logisland/master/logisland-framework/
↪logisland-resources/src/main/resources/conf/docker-compose.yml
docker-compose up
```

Note: you should add an entry for **sandbox** and **kafka** (with the container ip) in your `/etc/hosts` as it will be easier to access to all web services in logisland running container.

Any logisland script can now be launched by running a *logisland.sh* script within the logisland docker container like in the example below where we launch *index-apache-logs.yml* job :

```
docker exec -i -t logisland bin/logisland.sh --conf conf/index-apache-logs.yml
```

2. Through an Hadoop cluster (production way)

Now you have played with the tool, you're ready to deploy your jobs into a real distributed cluster. From an edge node of your cluster :

- download and extract the latest release of logisland
- export *SPARK_HOME* and *HADOOP_CONF_DIR* environment variables
- run *logisland.sh* launcher script with your job conf file.

```
cd /opt
sudo wget https://github.com/Hurence/logisland/releases/download/v1.1.2/logisland-1.1.
↪2-bin.tar.gz

export SPARK_HOME=/opt/spark-2.1.0-bin-hadoop2.7/
export HADOOP_CONF_DIR=$SPARK_HOME/conf

sudo /opt/logisland-1.1.2/bin/logisland.sh --conf /home/hurence/tom/logisland-conf/v0.
↪10.0/future-factory.yml
```

1.2.2 Run Logisland stream within Kubernetes : stage 1

This is the begining of a multiple part series of tutorials going through setting up a scalable Apache log indexation to Elasticsearch in kubernetes. This guide will bring you to a fully functionnal Kubernetes logisland setup.

Part 1 - Setting up Elasticsearch Part 2 - Setting up Kibana Part 3 - Setting up a three-node Zookeeper cluster Part 4 - Setting up a three-node Kafka cluster Part 5 - Setting up Logisland

Kafka and Zookeeper can be manually scaled up at any time by altering and re-applying configuration. Kubernetes also provides features for autoscaling, read more about auto scaling Kubernetes Pods should that be a requirement.

sources

- https://imti.co/kafka-kubernetes/
- https://github.com/kiritbasu/Fake-Apache-Log-Generator
- https://blog.gruntwork.io/automated-testing-for-kubernetes-and-helm-charts-using-terratest-a4ddc4e67344

0 - Pre-requisites & initial setup

First of all you'll need a Kubernetes cluster or a minikube cluster (https://kubernetes.io/docs/tasks/tools/install-minikube/). For the first option I would highly recommend to follow the Hello Minikube tutorial for those who don't have any background with Kubernetes. This will help to get minikube and kubectl commands installed. (Minikube is the local development Kubernetes environment and kubectl is the command line interface used to interact with Kubernetes cluster).

Shaving the Yak!

One or two commands that used in this post will be mac or linux specific. Reference this guide to get more up to date and OS specific commands. Once you've got the tools all installed, you can now follow along these steps to create a single node Elasticsearch cluster.

If you are using Minikube, make sure that its started properly by running this command

- for mac:

```
minikube start --vm-driver=hyperkit
```

- for linux (use virtualbox by default, so you have to install it) :

```
minikube start
```

Now set the Minikube context. The context is what determines which cluster kubectl is interacting with.

```
kubectl config use-context minikube
```

Verify that kubectl is configured to communicate with your cluster:

```
kubectl cluster-info
```

To view the nodes in the cluster, run

```
kubectl get nodes
```

Kubernetes Dashboard

Minikube includes the kubernetes dashboard as an addon which you can enable.

```
minikube addons list
```

returns

```
- default-storageclass: enabled
- coredns: disabled
- kube-dns: enabled
- ingress: disabled
- registry: disabled
- registry-creds: disabled
- addon-manager: enabled
- dashboard: enabled
- storage-provisioner: enabled
- heapster: disabled
- efk: disabled
```

You can enable an addon using:

```
minikube addons enable dashboard
```

You can then open the dashboard with command

```
minikube dashboard
```

Please note that on some virtual environments (like VirtualBox) the minikube VM may start with too few resources (you should allocate at least 4 CPUs and 6Go RAM)

Kubernetes setup

The best you can do is to follow the official guides to get the following tools up and running.

The Kubernetes command-line tool, **kubectl**, allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs. setup kubectl

Minikube, a tool that runs a single-node Kubernetes cluster in a virtual machine on your laptop is the easiest way to start with. setup minikube

---

Note: Deciding where to run Kubernetes depends on what resources you have available and how much flexibility you need. You can run Kubernetes almost anywhere, from your laptop to VMs on a cloud provider to a rack of bare metal servers. You can also set up a fully-managed cluster by running a single command or craft your own customized cluster on your bare metal servers. setup kubernetes

---

Namespace

In this guide, I use the fictional namespace *logisland*. You can create this namespace in your cluster or use your own.

Create the file *namespace.yml*:

```
apiVersion: v1
kind: Namespace
metadata:
  name: logisland
```

Apply the configuration:

```
kubectl create -f ./namespace.yml
```

If you wish to use your own namespace for this Kafka installation, be sure to replace *logisland* in the configurations below.

Persistent volumes

In Kubernetes, managing storage is a distinct problem from managing compute. The PersistentVolume subsystem provides an API for users and administrators that abstracts details of how storage is provided from how it is consumed. To do this we introduce two new API resources: PersistentVolume and PersistentVolumeClaim.

A **PersistentVolume (PV)** is a piece of storage in the cluster that has been provisioned by an administrator. It is a resource in the cluster just like a node is a cluster resource. PVs are volume plugins like Volumes, but have a lifecycle independent of any individual pod that uses the PV. This API object captures the details of the implementation of the storage,

be that NFS, iSCSI, or a cloud-provider-specific storage system.

A **PersistentVolumeClaim (PVC)** is a request for storage by a user. It is similar to a pod. Pods consume node resources and PVCs consume PV resources. Pods can request specific levels of resources (CPU and Memory). Claims can request specific size and access modes (e.g., can be mounted once read/write or many times read-only).

Create the local folders where you want to store your files (change this to wherever you want to store data on your nodes) :

```
mkdir /tmp/data
```

Create the file *pv-volume.yml*

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: datadir
  labels:
    app: kafka
    type: local
  namespace: logisland
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/tmp/data"
```

Apply the configuration:

```
kubectl create -f ./pv-volume.yml
```

Configuration maps

We will need a few configuration variables in our setup to bind containers together and define some environment variables. The first config map is specific to *loggen* tool which is a wrapped python program that sends fake generated apache logs to a given Kafka topic at a specified rate. The second one is a set of settings that will be used by the *logisland* job in order to configure itself. We'll go into deeper details in the last section of this post.

Create the file *config-maps.yml* with the following content

```
apiVersion: v1
kind: ConfigMap
metadata:
```

(continues on next page)

(continued from previous page)

```
  name: special-config
  namespace: logisland
data:
  loggen.sleep: '0.2'
  loggen.num: '0'
  loggen.topic: logisland_raw
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: logisland-config
  namespace: logisland
data:
  kafka.brokers: kafka:9092
  zk.quorum: zookeeper:2181
  es.hosts: elasticsearch:9300
  es.cluster.name: es-logisland
```

Apply the configuration:

```
kubectl create -f ./config-maps.yml
```

1 - Setting up Elasticsearch cluster on Kubernetes

Single Node Elasticsearch Cluster

Create the file *elasticsearch-service.yml*:

```
apiVersion: v1
kind: Service
metadata:
  name: elasticsearch
  namespace: logisland
  labels:
    component: elasticsearch
spec:
  type: ClusterIP
  selector:
    component: elasticsearch
  ports:
    - name: http
      port: 9200
      protocol: TCP
    - name: tcp
      port: 9300
      protocol: TCP
```

Apply the configuration:

```
kubectl create -f ./elasticsearch-service.yml
```

Create the file *elasticsearch-deployment.yml*:

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: elasticsearch
  namespace: logisland
spec:
  selector:
    matchLabels:
      component: elasticsearch
  template:
    metadata:
      labels:
        component: elasticsearch
    spec:
      containers:
        - name: elasticsearch
          image: docker.elastic.co/elasticsearch/elasticsearch:5.4.3
          env:
            - name: discovery.type
              value: single-node
            - name: cluster.name
              value: "es-logisland"
            - name: xpack.security.enabled
              value: "false"
          ports:
            - containerPort: 9200
              name: http
              protocol: TCP
            - containerPort: 9300
              name: tcp
              protocol: TCP
```

Apply the configuration:

```
kubectl create -f ./elasticsearch-deployment.yml
```

Expose the cluster

We can verify that the cluster is running by looking at the logs. But, let's check if elasticsearch api is responding first.

In a seperate shell window, excute the following to start a proxy into Kubernetes cluster.

```
kubectl -n logisland port-forward svc/elasticsearch 9200:9200
```

Now, back in the other window, lets execute a curl command to get the response from the pod via the proxy.

```
curl http://localhost:9200
```

Outputs:

```
{
  "name" : "19SlwE4",
  "cluster_name" : "es-logisland",
  "cluster_uuid" : "ef41SIbWRHmSDoDhcFA9WA",
  "version" : {
    "number" : "5.4.3",
```

(continues on next page)

(continued from previous page)

```
    "build_hash" : "eed30a8",
    "build_date" : "2017-06-22T00:34:03.743Z",
    "build_snapshot" : false,
    "lucene_version" : "6.5.1"
  },
  "tagline" : "You Know, for Search"
}
```

Great, everything is working.

2 - Setup Kibana

Let's try to setup kibana pointing to our elasticsearch single node cluster.

Create the file *kibana-service.yml*:

```
apiVersion: v1
kind: Service
metadata:
  name: kibana
  namespace: logisland
  labels:
    component: kibana
spec:
  type: NodePort
  selector:
    component: kibana
  ports:
    - name: http
      port: 5601
      targetPort: 5601
      nodePort: 30123
      protocol: TCP
```

Apply the configuration:

```
kubectl create -f ./kibana-service.yml
```

Create the file *kibana-deployment.yml*:

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: kibana
  namespace: logisland
spec:
  selector:
    matchLabels:
      component: kibana
  template:
    metadata:
      labels:
        component: kibana
    spec:
      containers:
```

(continues on next page)

(continued from previous page)

```
        - name: kibana
          image: docker.elastic.co/kibana/kibana:5.4.3
          env:
            - name: ELASTICSEARCH_URL
              value: http://elasticsearch:9200
            - name: XPACK_SECURITY_ENABLED
              value: "true"
          ports:
            - containerPort: 5601
              name: http
              protocol: TCP
```

Apply the configuration:

```
kubectl create -f ./kibana-deployment.yml
```

To access kibana through your localhost forward the port

```
kubectl -n logisland port-forward svc/kibana 5601:5601
```

3 - Setting up Zookeeper

Kafka requires Zookeeper for maintaining configuration information, naming, providing distributed synchronization, and providing group services to coordinate its nodes.

Zookeeper Headless Service

Kubernetes Services are persistent and provide a stable and reliable way to connect to Pods.

Setup a Kubernetes Service named kafka-zookeeper in namespace *logisland*. The kafka-zookeeper service resolves the domain name kafka-zookeeper to an internal ClusterIP. The automatically assigned ClusterIP uses Kubernetes internal proxy to load balance calls to any Pods found from the configured selector, in this case, app: kafka-zookeeper.

After setting up the kafka-zookeeper Service, a DNS lookup from within the cluster may produce a result similar to the following:

```
# nslookup kafka-zookeeper
Server:         10.96.0.10
Address:    10.96.0.10#53

Name:    kafka-zookeeper.logisland.svc.cluster.local
Address: 10.103.184.71
```

In the example above, 10.103.184.71 is the internal IP address of the ** kafka-zookeeper* service itself and proxies calls to one of the Zookeeper Pods it finds labeled app: kafka-zookeeper. At this point, no Pods are available until added further down. However, the service finds them when they become active.

Create the file *zookeeper-service.yml*:

```
apiVersion: v1
kind: Service
metadata:
  name: kafka-zookeeper
  namespace: logisland
```

(continues on next page)

(continued from previous page)

```
spec:
  ports:
    - name: client
      port: 2181
      protocol: TCP
      targetPort: client
  selector:
    app: kafka-zookeeper
  sessionAffinity: None
  type: ClusterIP
```

Apply the configuration:

```
kubectl create -f ./zookeeper-service.yml
```

Zookeeper Headless Service

A Kubernetes Headless Service does not resolve to a single IP; instead, Headless Services returns the IP addresses of any Pods found by their selector, in this case, Pods labeled app: kafka-zookeeper.

Once Pods labeled app: kafka-zookeeper are running, this Headless Service returns the results of an in-cluster DNS lookup similar to the following:

```
# nslookup kafka-zookeeper
Server:        10.96.0.10
Address:    10.96.0.10#53

Name:    kafka-zookeeper-headless.logisland.svc.cluster.local
Address: 192.168.108.150
Name:    kafka-zookeeper-headless.logisland.svc.cluster.local
Address: 192.168.108.181
Name:    kafka-zookeeper-headless.logisland.svc.cluster.local
Address: 192.168.108.132
```

In the example above, the Kubernetes Service kafka-zookeeper-headless returned the internal IP addresses of three individual Pods.

At this point, no Pod IPs can be returned until the Pods are configured in the StatefulSet further down.

Create the file *zookeeper-service-headless.yml*:

```
apiVersion: v1
kind: Service
metadata:
  name: kafka-zookeeper-headless
  namespace: logisland
spec:
  #clusterIP: None
  ports:
    - name: client
      port: 2181
      protocol: TCP
      targetPort: 2181
    - name: election
      port: 3888
      protocol: TCP
```

(continues on next page)

(continued from previous page)

```
      targetPort: 3888
    - name: server
      port: 2888
      protocol: TCP
      targetPort: 2888
  selector:
    app: kafka-zookeeper
  sessionAffinity: None
  type: ClusterIP
```

Apply the configuration:

```
kubectl create -f ./zookeeper-service-headless.yml
```

Zookeeper StatefulSet

Kubernetes StatefulSets offer stable and unique network identifiers, persistent storage, ordered deployments, scaling, deletion, termination, and automated rolling updates.

Unique network identifiers and persistent storage are essential for stateful cluster nodes in systems like Zookeeper and Kafka. While it seems strange to have a coordinator like Zookeeper running inside a Kubernetes cluster sitting on its own coordinator Etcd, it makes sense since these systems are built to run independently. Kubernetes supports running services like Zookeeper and Kafka with features like headless services and stateful sets which demonstrates the flexibility of Kubernetes as both a microservices platform and a type of virtual infrastructure.

The following configuration creates three kafka-zookeeper Pods, kafka-zookeeper-0, kafka-zookeeper-1, kafka-zookeeper-2 and can be scaled to as many as desired. Ensure that the number of specified replicas matches the environment variable ZK_REPLICAS specified in the container spec.

Pods in this StatefulSet run the Zookeeper Docker image gcr.io/google_samples/k8szk:v3, which is a sample image provided by Google for testing GKE, it is recommended to use custom and maintained Zookeeper image once you are familiar with this setup.

Create the file *zookeeper-statefulset.yml*:

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: kafka-zookeeper
  namespace: logisland
spec:
  podManagementPolicy: OrderedReady
  replicas: 3
  revisionHistoryLimit: 1
  selector:
    matchLabels:
      app: kafka-zookeeper
  serviceName: kafka-zookeeper-headless
  template:
    metadata:
      labels:
        app: kafka-zookeeper
    spec:
      containers:
        - command:
            - /bin/bash
```

(continues on next page)

(continued from previous page)

```
        - -xec
        - zkGenConfig.sh && exec zkServer.sh start-foreground
      env:
        - name: ZK_REPLICAS
          value: "3"
        - name: JMXAUTH
          value: "false"
        - name: JMXDISABLE
          value: "false"
        - name: JMXPORT
          value: "1099"
        - name: JMXSSL
          value: "false"
        - name: ZK_CLIENT_PORT
          value: "2181"
        - name: ZK_ELECTION_PORT
          value: "3888"
        - name: ZK_HEAP_SIZE
          value: 1G
        - name: ZK_INIT_LIMIT
          value: "5"
        - name: ZK_LOG_LEVEL
          value: INFO
        - name: ZK_MAX_CLIENT_CNXNS
          value: "60"
        - name: ZK_MAX_SESSION_TIMEOUT
          value: "40000"
        - name: ZK_MIN_SESSION_TIMEOUT
          value: "4000"
        - name: ZK_PURGE_INTERVAL
          value: "0"
        - name: ZK_SERVER_PORT
          value: "2888"
        - name: ZK_SNAP_RETAIN_COUNT
          value: "3"
        - name: ZK_SYNC_LIMIT
          value: "10"
        - name: ZK_TICK_TIME
          value: "2000"
      image: gcr.io/google_samples/k8szk:v3
      imagePullPolicy: IfNotPresent
      livenessProbe:
        exec:
          command:
            - zkOk.sh
        failureThreshold: 3
        initialDelaySeconds: 20
        periodSeconds: 10
        successThreshold: 1
        timeoutSeconds: 1
      name: zookeeper
      ports:
        - containerPort: 2181
          name: client
          protocol: TCP
        - containerPort: 3888
          name: election
```

(continues on next page)

(continued from previous page)

```
            protocol: TCP
          - containerPort: 2888
            name: server
            protocol: TCP
        readinessProbe:
          exec:
            command:
              - zkOk.sh
          failureThreshold: 3
          initialDelaySeconds: 20
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 1
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
        volumeMounts:
          - mountPath: /var/lib/zookeeper
            name: data
    dnsPolicy: ClusterFirst
    restartPolicy: Always
    schedulerName: default-scheduler
    securityContext:
      fsGroup: 1000
      runAsUser: 1000
    terminationGracePeriodSeconds: 30
    volumes:
      - emptyDir: {}
        name: data
  updateStrategy:
    type: OnDelete
```

Apply the configuration:

```
kubectl create -f ./zookeeper-statefulset.yml
```

Zookeeper PodDisruptionBudget

PodDisruptionBudget can help keep the Zookeeper service stable during Kubernetes administrative events such as draining a node or updating Pods.

From the official documentation for PDB (PodDisruptionBudget):

A PDB specifies the number of replicas that an application can tolerate having, relative to how many it is intended to have. For example, a Deployment which has a .spec.replicas: 5 is supposed to have 5 pods at any given time. If its PDB allows for there to be 4 at a time, then the Eviction API will allow voluntary disruption of one, but not two pods, at a time.

The configuration below tells Kubernetes that we can only tolerate one of our Zookeeper Pods down at any given time. maxUnavailable may be set to a higher number if we increase the number of Zookeeper Pods in the StatefulSet.

Create the file *zookeeper-disruptionbudget.yml*:

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
```

(continues on next page)

(continued from previous page)

```
  labels:
    app: kafka-zookeeper
  name: kafka-zookeeper
  namespace: logisland
spec:
  maxUnavailable: 1
  selector:
    matchLabels:
      app: kafka-zookeeper
```

Apply the configuration:

```
kubectl create -f ./zookeeper-disruptionbudget.yml
```

4 - Setting up Kafka

Once Zookeeper is up and running we have satisfied the requirements for Kafka. Kafka is set up in a similar configuration to Zookeeper, utilizing a Service, Headless Service and a StatefulSet.

Kafka Service

The following Service provides a persistent internal Cluster IP address that proxies and load balance requests to Kafka Pods found with the label app: kafka and exposing the port 9092.

Create the file *kafka-service.yml*:

apiVersion: v1 kind: Service metadata:

name: kafka namespace: logisland

spec:

ports:

- name: broker port: 9092 protocol: TCP targetPort: kafka

selector: app: kafka

sessionAffinity: None type: ClusterIP

Apply the configuration:

```
kubectl create -f ./kafka-service.yml
```

Kafka Headless Service

The following Headless Service provides a list of Pods and their internal IPs found with the label app: kafka and exposing the port 9092. The previously created Service: kafka always returns a persistent IP assigned at the creation time of the Service. The following kafka-headless services return the domain names and IP address of individual Pods and are liable to change as Pods are added, removed or updated.

Create the file *kafka-service-headless.yml*:

apiVersion: v1 kind: Service metadata:

name: kafka-headless namespace: logisland

spec: #clusterIP: None ports:

- name: broker port: 9092 protocol: TCP targetPort: 9092

selector: app: kafka

sessionAffinity: None type: ClusterIP

Apply the configuration:

```
kubectl create -f ./kafka-service-headless.yml
```

Kafka StatefulSet

The following StatefulSet deploys Pods running the confluentinc/cp-kafka:4.1.2-2 Docker image from Confluent.

Each pod is assigned 1Gi of storage using the rook-block storage class. See Rook.io for more information on file, block, and object storage services for cloud-native environments.

Create the file *kafka-statefulset.yml*:

apiVersion: apps/v1 kind: StatefulSet metadata:

labels: app: kafka

name: kafka namespace: logisland

spec: podManagementPolicy: OrderedReady replicas: 3 revisionHistoryLimit: 1 selector:

matchLabels: app: kafka

serviceName: kafka-headless template:

metadata:

labels: app: kafka

spec:

containers:

- **command:**
  - sh
  - -exc
  - unset KAFKA_PORT && export KAFKA_BROKER_ID=${HOSTNAME##*-} && export KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://${POD_IP}:9092 && exec /etc/confluent/docker/run

env:

- name: POD_IP valueFrom:

fieldRef: apiVersion: v1 fieldPath: status.podIP

- name: KAFKA_HEAP_OPTS value: -Xmx1G -Xms1G
- name: KAFKA_ZOOKEEPER_CONNECT value: kafka-zookeeper:2181

value: 10.105.213.202:2181 # value: ${KAFKA_ZOOKEEPER_SERVICE_HOST}:2181

- name: KAFKA_LOG_DIRS value: /opt/kafka/data/logs

– name: KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR value: "3"

– name: KAFKA_JMX_PORT value: "5555"

image: confluentinc/cp-kafka:4.1.2-2 imagePullPolicy: IfNotPresent livenessProbe:

exec:

command:

– sh

– -ec

– /usr/bin/jps | /bin/grep -q SupportedKafka

failureThreshold: 3 initialDelaySeconds: 30 periodSeconds: 10 successThreshold: 1 timeoutSeconds: 5

name: kafka-broker ports:

– containerPort: 9092 name: kafka protocol: TCP

readinessProbe: failureThreshold: 3 initialDelaySeconds: 30 periodSeconds: 10 successThreshold: 1 tcpSocket:

port: kafka

timeoutSeconds: 5

resources: {} terminationMessagePath: /dev/termination-log terminationMessagePolicy: File volumeMounts:

– mountPath: /opt/kafka/data name: datadir-claim

dnsPolicy: ClusterFirst restartPolicy: Always schedulerName: default-scheduler securityContext: {} terminationGracePeriodSeconds: 60

updateStrategy: type: OnDelete

volumeClaimTemplates:

- **metadata:** name: datadir-claim

  spec: #storageClassName: "standard" # storageClassName: rook-block accessModes:

  – ReadWriteOnce

  resources:

  requests: storage: 1Gi

Apply the configuration:

```
kubectl create -f ./kafka-statefulset.yml
```

Kafka Test Pod

Add a test Pod to help explore and debug your new Kafka cluster. The Confluent Docker image confluentinc/cp-kafka:4.1.2-2 used for the test Pod is the same as our nodes from the StatefulSet and contain useful command in the /usr/bin/ folder.

Create the file kafka-test.yml:

apiVersion: v1 kind: Pod metadata:

name: kafka-test-client namespace: logisland

spec:

containers:

- **command:**
  - sh
  - -c
  - exec tail -f /dev/null

image: confluentinc/cp-kafka:4.1.2-2 imagePullPolicy: IfNotPresent name: kafka resources: {} terminationMessagePath: /dev/termination-log terminationMessagePolicy: File

Apply the configuration:

```
kubectl create -f ./kafka-test.yml
```

5 - Working with Kafka

If you have deployed the kafka-test-client pod from the configuration above, the following commands should get you started with some basic operations:

Create Topic

```
kubectl -n logisland exec kafka-test-client -- \
/usr/bin/kafka-topics --zookeeper kafka-zookeeper:2181 \
--topic logisland_raw --create --partitions 3 --replication-factor 1
```

List Topics

```
kubectl -n logisland exec kafka-test-client -- \
```

/usr/bin/kafka-topics –zookeeper kafka-zookeeper:2181 –list

Sending logs to Kafka

This script generates a boatload of fake apache logs very quickly. Its useful for generating fake workloads for data ingest and/or analytics applications. It can write log lines to console, to log files or directly to gzip files. Or to kafka … It utilizes the excellent Faker library to generate realistic ip's, URI's etc.

Create the file *loggen-deployment.yml*:

apiVersion: v1 kind: Pod metadata:

name: loggen-job namespace: logisland

spec:

containers:

- name: loggen image: hurence/loggen imagePullPolicy: IfNotPresent env:
  - name: LOGGEN_SLEEP valueFrom:

    configMapKeyRef: name: special-config key: loggen.sleep
  - name: LOGGEN_NUM valueFrom:

    configMapKeyRef: name: special-config key: loggen.num
  - name: LOGGEN_KAFKA valueFrom:

    configMapKeyRef: name: logisland-config key: kafka.brokers
  - name: LOGGEN_KAFKA_TOPIC valueFrom:

    configMapKeyRef: name: special-config key: loggen.topic

Apply the configuration:

```
kubectl create -f ./loggen-deployment.yml
```

Listen on a Topic

make sure some fake apache logs are flowing through kafka topic

```
kubectl -n logisland exec -ti kafka-test-client -- \
/usr/bin/kafka-console-consumer --bootstrap-server kafka:9092 \
--topic logisland_raw --from-beginning
```

6 - Setup logisland

It's now time time to dive into log mining. We'll setup a 3 instances logisland stream that will handle apache logs parsing (coming from loggen script) as a ReplicaSet

Create the file *logisland-deployment.yml*:

```
apiVersion: apps/v1beta2
kind: ReplicaSet
metadata:
  name: logisland-job
  namespace: logisland
spec:
  replicas: 3
  selector:
    matchLabels:
      app: logisland-job
  template:
    metadata:
      labels:
        app: logisland-job
    spec:
      containers:
        - name: logisland
          image: hurence/logisland-job
          imagePullPolicy: IfNotPresent
          command: ["/opt/logisland/bin/logisland.sh"]
          args: ["--standalone", "--conf", "/opt/logisland/conf/index-apache-logs-
↪plainjava.yml"]
```

(continues on next page)

(continued from previous page)

```
          env:
            - name: ES_CLUSTER_NAME
              valueFrom:
                configMapKeyRef:
                  name: logisland-config
                  key: es.cluster.name
            - name: KAFKA_BROKERS
              valueFrom:
                configMapKeyRef:
                  name: logisland-config
                  key: kafka.brokers
            - name: ES_HOSTS
              valueFrom:
                configMapKeyRef:
                  name: logisland-config
                  key: es.hosts
```

Apply the configuration:

```
kubectl create -f ./logisland-deployment.yml
```

run the following command to see events parsed by logisland flowing through the output topic

> kubectl -n logisland exec -ti kafka-test-client – /usr/bin/kafka-console-consumer –bootstrap-server kafka:9092 –topic logisland_events

check that logs are correctly stored into elasticsearch

> kubectl -n logisland exec -ti kafka-test-client – curl http://elasticsearch:9200/logisland.*/_search?pretty=1

1.2.3 Apache logs indexing

In the following getting started tutorial we'll drive you through the process of Apache log mining with LogIsland platform.

Note: It is possible to store data in different datastores. In this tutorial, we will see the case of ElasticSearch ,Solr and MongoDb.

- Apache logs indexing into elasticsearch
- Apache logs indexing into solr
- Apache logs indexing into mongodb

1.2.4 Apache logs indexing with elasticsearch

In the following getting started tutorial we'll drive you through the process of Apache log mining with LogIsland platform. The final data will be stored in elasticsearch

This tutorial is very similar to :

- Apache logs indexing into solr
- Apache logs indexing into mongodb

Note: Please note that you should not launch silmutaneously several docker-compose because we are exposing local port in them. So running several at the same time would be conflicting. So be sure to have killed all your currently running containers.

1.Install required components

- You either use docker-compose with available docker-compose-index-apache-logs-es.yml file in the tar.gz assembly in the conf folder.

  In this case you can skip this section

- Or you can launch the job in your cluster, but in this case you will have to make changes to job conf file so it works in your environment.

  In this case please make sure to already have installed elasticsearch modules (depending on which base you will use).

If not you can just do it through the components.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_6_6_2-
↪client:1.1.0
```

Note: In the following sections we will use docker-compose to run the job. (please install it before pursuing if you are not using your own cluster)

2. Logisland job setup

The logisland job that we will use is **./conf/index-apache-logs-es.yml** The logisland docker-compose file that we will use is **./conf/docker-compose-index-apache-logs-es.yml**

We will start by explaining each part of the config file.

An Engine is needed to handle the stream processing. This `conf/index-apache-logs-es.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode with 2 cpu cores and 2G of RAM.

```
engine:
  component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
  type: engine
  documentation: Index some apache logs with logisland
  configuration:
    spark.app.name: IndexApacheLogsDemo
    spark.master: local[2]
    spark.driver.memory: 1G
    spark.driver.cores: 1
    spark.executor.memory: 2G
    spark.executor.instances: 4
    spark.executor.cores: 2
    spark.yarn.queue: default
    spark.yarn.maxAppAttempts: 4
    spark.yarn.am.attemptFailuresValidityInterval: 1h
    spark.yarn.max.executor.failures: 20
    spark.yarn.executor.failuresValidityInterval: 1h
```

(continues on next page)

(continued from previous page)

```
    spark.task.maxFailures: 8
    spark.serializer: org.apache.spark.serializer.KryoSerializer
    spark.streaming.batchDuration: 1000
    spark.streaming.backpressure.enabled: false
    spark.streaming.unpersist: false
    spark.streaming.blockInterval: 500
    spark.streaming.kafka.maxRatePerPartition: 3000
    spark.streaming.timeout: -1
    spark.streaming.kafka.maxRetries: 3
    spark.streaming.ui.retainedBatches: 200
    spark.streaming.receiver.writeAheadLog.enable: false
    spark.ui.port: 4050
```

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job, here an Elasticsearch service that will be used later in the `BulkAddElasticsearch` processor.

```
- controllerService: elasticsearch_service
  component: com.hurence.logisland.service.elasticsearch.Elasticsearch_5_4_0_
↪ClientService
  type: service
  documentation: elasticsearch service
  configuration:
    hosts: ${ES_HOSTS}
    cluster.name: ${ES_CLUSTER_NAME}
    batch.size: 5000
```

Note: As you can see it uses environment variable so make sure to set them. (if you use the docker-compose file of this tutorial it is already done for you)

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our ouput records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that converts raw apache logs into structured log records
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: ${KAFKA_BROKERS}
    kafka.zookeeper.quorum: ${ZK_QUORUM}
    kafka.topic.autoCreate: true
```

(continues on next page)

(continued from previous page)

```
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

Note: As you can see it uses environment variable so make sure to set them. (if you use the docker-compose file of this tutorial it is already done for you)

Within this stream a `SplitText` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
# parse apache logs into logisland records
- processor: apache_parser
  component: com.hurence.logisland.processor.SplitText
  type: parser
  documentation: a parser that produce events from an apache log REGEX
  configuration:
    record.type: apache_log
    value.regex: (\S+)\s+(\S+)\s+(\S+)\s+\[([\w:\/]+\s[+\-]\d{4})\]\s+
↪"(\S+)\s+(\S+)\s*(\S*)"\s+(\S+)\s+(\S+)
    value.fields: src_ip,identd,user,record_time,http_method,http_query,http_version,
↪http_status,bytes_out
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

The second processor will handle `Records` produced by the `SplitText` to index them into elasticsearch

```
# all the parsed records are added to elasticsearch by bulk
- processor: es_publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: a processor that indexes processed events in elasticsearch
  configuration:
    elasticsearch.client.service: elasticsearch_service
    default.index: logisland
    default.type: event
    timebased.index: yesterday
    es.index.field: search_index
    es.type.field: record_type
```

3. Launch the job

For this tutorial we will handle some apache logs with a splitText parser and send them to Elastiscearch. Launch your docker container with this command (we suppose you are in the root of the tar gz assembly) :

```
sudo docker-compose -f ./conf/docker-compose-index-apache-logs-es.yml up -d
```

Make sure all container are running and that there is no error.

```
sudo docker-compose ps
```

Those containers should be visible and running

‘‘‘ CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES 0d9e02b22c38 docker.elastic.co/kibana/kibana:5.4.0 “/bin/sh -c /usr/loc…” 13 seconds ago Up 8 seconds 0.0.0.0:5601->5601/tcp

conf_kibana_1 ab15f4b5198c docker.elastic.co/elasticsearch/elasticsearch:6.6.2 “/bin/bash bin/es-do…” 13 seconds ago Up 7 seconds 0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp conf_elasticsearch_1 a697e45d2d1a hurence/logisland:1.1.0 “tail -f bin/logisla…” 13 seconds ago Up 9 seconds 0.0.0.0:4050->4050/tcp, 0.0.0.0:8082->8082/tcp, 0.0.0.0:9999->9999/tcp conf_logisland_1 db80cdf23b45 hurence/zookeeper “/bin/sh -c ‘/usr/sb…” 13 seconds ago Up 10 seconds 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, 7072/tcp conf_zookeeper_1 7aa7a87dd16b hurence/kafka:0.10.2.2-scala-2.11 “start-kafka.sh” 13 seconds ago Up 5 seconds 0.0.0.0:9092->9092/tcp conf_kafka_1

“‘

```
sudo docker logs conf_kibana_1
sudo docker logs conf_elasticsearch_1
sudo docker logs conf_logisland_1
sudo docker logs conf_zookeeper_1
sudo docker logs conf_kafka_1
```

Should not return errors or any suspicious messages

you can now run the job inside the logisland container

```
sudo docker exec -ti conf_logisland_1 ./bin/logisland.sh --conf ./conf/index-apache-
↪logs-es.yml
```

The last logs should be something like :

2019-03-19 16:08:47 INFO StreamProcessingRunner:95 - awaitTermination for engine 1 2019-03-19 16:08:47 WARN SparkContext:66 - Using an existing SparkContext; some configuration may not take effect.

4. Inject some Apache logs into the system

Now we’re going to send some logs to `logisland_raw` Kafka topic.

If you don’t have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

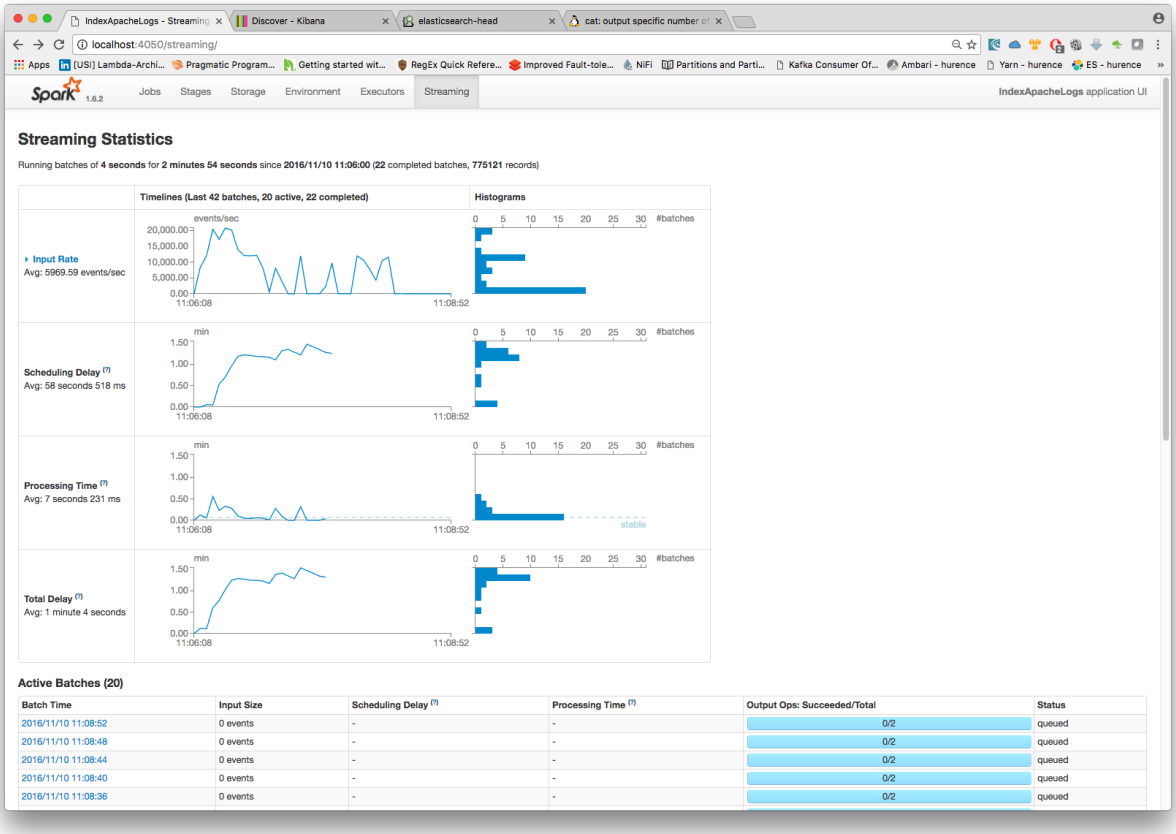
Let’s send the first 500 lines of NASA http access over July 1995 to LogIsland with kafka scripts (available in our logisland container) to `logisland_raw` Kafka topic.

In another terminal run those commands

```
sudo docker exec -ti conf_logisland_1 bash
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -n 500 NASA_access_log_Jul95 | ${KAFKA_HOME}/bin/kafka-console-producer.sh --
↪broker-list kafka:9092 --topic logisland_raw
```

5. Monitor your spark jobs and Kafka topics

Now go to http://localhost:4050/streaming/ to see how fast Spark can process your data



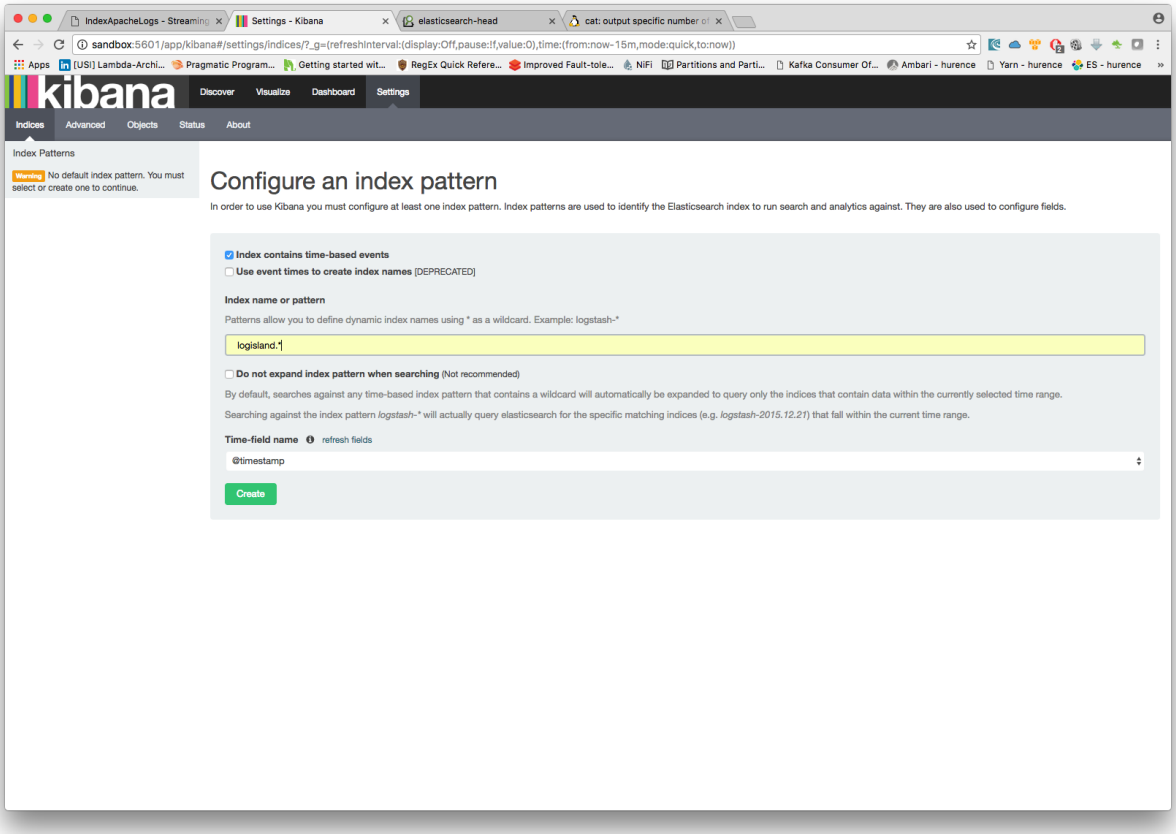
6. Inspect the logs

Kibana

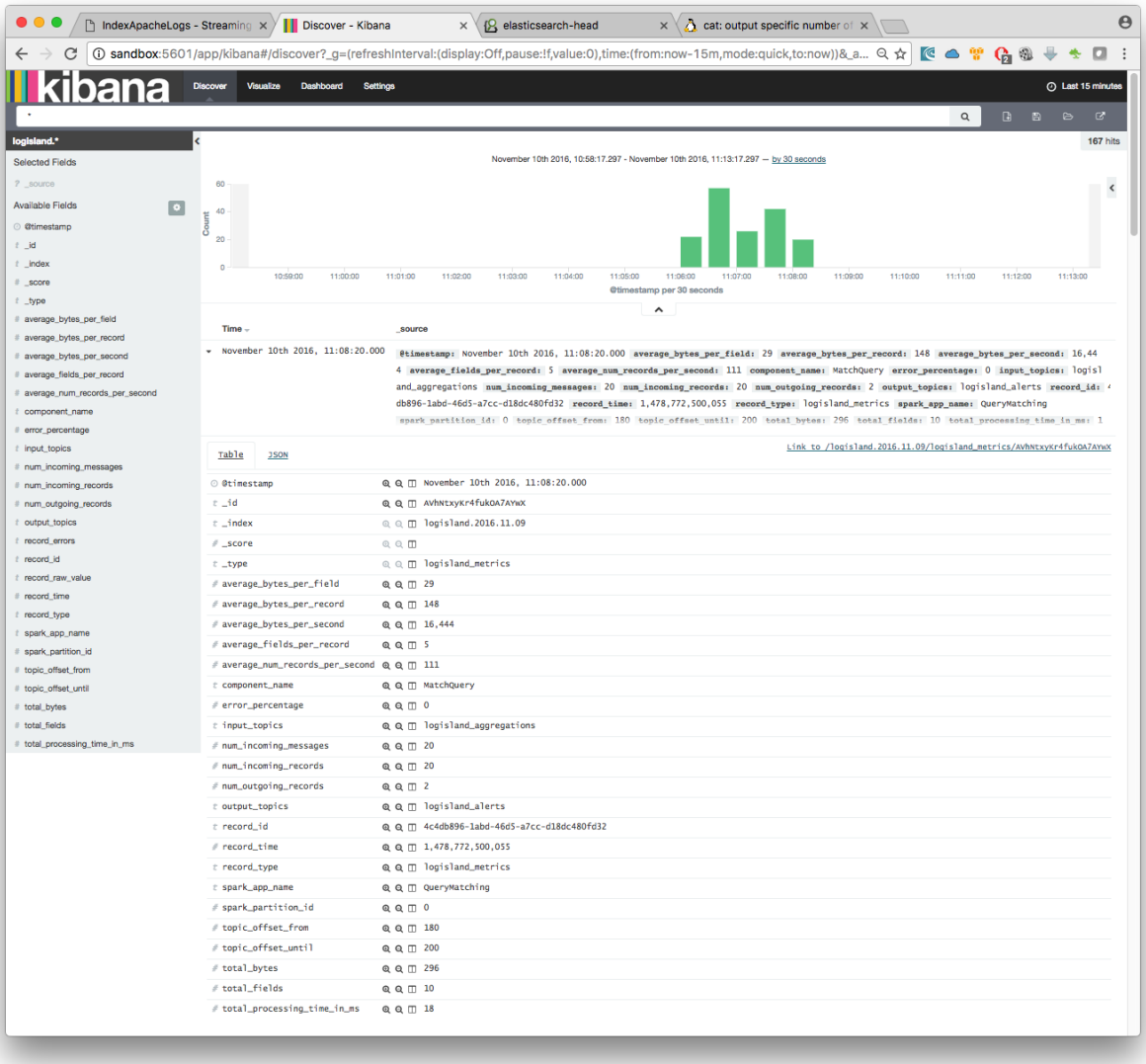
With ElasticSearch, you can use Kibana. We included one in our docker-compose file.

Open up your browser and go to http://localhost:5601/ and you should be able to explore your apache logs.

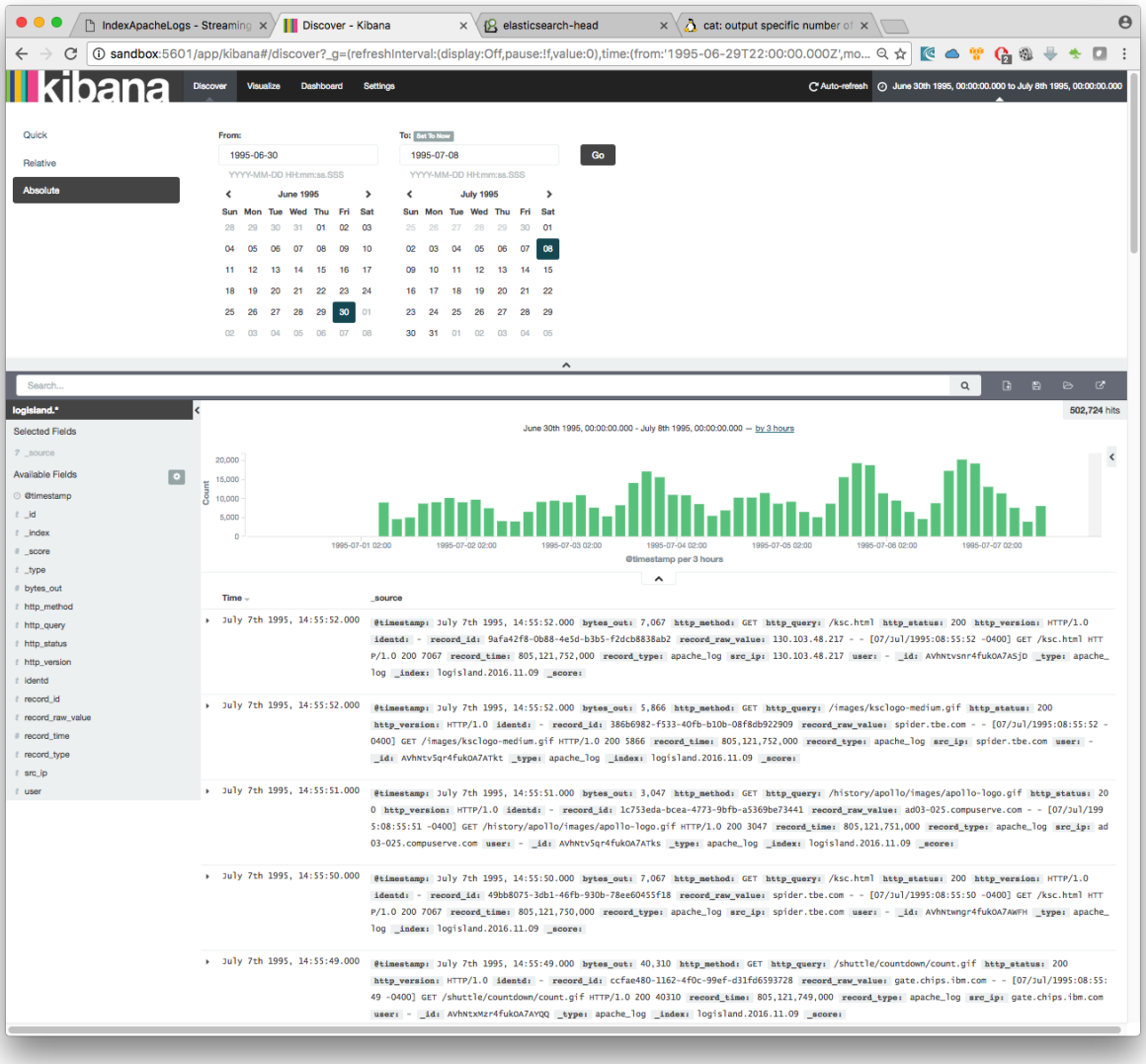
Configure a new index pattern with `logisland.*` as the pattern name and `@timestamp` as the time value field.



Then if you go to Explore panel for the latest 15' time window you'll only see logisland process_metrics events which give you insights about the processing bandwidth of your streams.



As we explore data logs from july 1995 we'll have to select an absolute time filter from 1995-06-30 to 1995-07-08 to see the events.



3. Stop the job

You can Ctr+c the console where you launched logisland job. Then to kill all containers used run :

```
sudo docker-compose -f ./conf/docker-compose-index-apache-logs-es.yml down
```

Make sure all container have disappeared.

```
sudo docker ps
```

1.2.5 Apache logs indexing with mongo

In the following getting started tutorial we'll drive you through the process of Apache log mining with LogIsland platform. The final data will be stored in mongo

This tutorial is very similar to :

- Apache logs indexing into solr
- Apache logs indexing into elasticsearch

Note: Please note that you should not launch silmutaneously several docker-compose because we are exposing local port in them. So running several at the same time would be conflicting. So be sure to have killed all your currently running containers.

1.Install required components

- You either use docker-compose with available docker-compose-index-apache-logs-mongo.yml file in the tar.gz assembly in the conf folder.

  In this case you can skip this section

- Or you can launch the job in your cluster, but in this case you will have to make changes to job conf file so it works in your environment.

  In this case please make sure to already have installed mongo modules (depending on which base you will use).

If not you can just do it through the components.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-service-mongodb-client:1.1.2
```

Note: In the following sections we will use docker-compose to run the job. (please install it before pursuing if you are not using your own cluster)

2. Logisland job setup

The logisland job that we will use is **./conf/index-apache-logs-mongo.yml** The logisland docker-compose file that we will use is **./conf/docker-compose-index-apache-logs-mongo.yml**

We will start by explaining each part of the config file.

An Engine is needed to handle the stream processing. This `conf/index-apache-logs-mongo.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode with 2 cpu cores and 2G of RAM.

```
engine:
  component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
  type: engine
  documentation: Index some apache logs with logisland
  configuration:
    spark.app.name: IndexApacheLogsDemo
    spark.master: local[2]
    spark.driver.memory: 1G
    spark.driver.cores: 1
    spark.executor.memory: 2G
    spark.executor.instances: 4
    spark.executor.cores: 2
    spark.yarn.queue: default
    spark.yarn.maxAppAttempts: 4
```

(continues on next page)

(continued from previous page)

```
    spark.yarn.am.attemptFailuresValidityInterval: 1h
    spark.yarn.max.executor.failures: 20
    spark.yarn.executor.failuresValidityInterval: 1h
    spark.task.maxFailures: 8
    spark.serializer: org.apache.spark.serializer.KryoSerializer
    spark.streaming.batchDuration: 1000
    spark.streaming.backpressure.enabled: false
    spark.streaming.unpersist: false
    spark.streaming.blockInterval: 500
    spark.streaming.kafka.maxRatePerPartition: 3000
    spark.streaming.timeout: -1
    spark.streaming.kafka.maxRetries: 3
    spark.streaming.ui.retainedBatches: 200
    spark.streaming.receiver.writeAheadLog.enable: false
    spark.ui.port: 4050
```

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job, here an mongo service that will be used later in the `TODO` processor.

```
- controllerService: datastore_service
  component: com.hurence.logisland.service.mongodb.MongoDBControllerService
  type: service
  documentation: "Mongo 3.8.0 service"
  configuration:
    mongo.uri: ${MONGO_URI}
    mongo.db.name: logisland
    mongo.collection.name: apache
    # possible values ACKNOWLEDGED, UNACKNOWLEDGED, FSYNCED, JOURNALED, REPLICA_
↪ACKNOWLEDGED, MAJORITY
    mongo.write.concern: ACKNOWLEDGED
    flush.interval: 2000
    batch.size: 100
```

Note: As you can see it uses environment variable so make sure to set them. (if you use the docker-compose file of this tutorial it is already done for you)

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our ouput records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that converts raw apache logs into structured log records
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
```

(continues on next page)

(continued from previous page)

```
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: ${KAFKA_BROKERS}
    kafka.zookeeper.quorum: ${ZK_QUORUM}
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

Note: As you can see it uses environment variable so make sure to set them. (if you use the docker-compose file of this tutorial it is already done for you)

Within this stream a `SplitText` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
# parse apache logs into logisland records
- processor: apache_parser
  component: com.hurence.logisland.processor.SplitText
  type: parser
  documentation: a parser that produce events from an apache log REGEX
  configuration:
    record.type: apache_log
    value.regex: (\S+)\s+(\S+)\s+(\S+)\s+\[([\w:\/]+\s[+\-]\d{4})\]\s+
↪"(\S+)\s+(\S+)\s*(\S*)"\s+(\S+)\s+(\S+)
    value.fields: src_ip,identd,user,record_time,http_method,http_query,http_version,
↪http_status,bytes_out
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

The second processor will handle `Records` produced by the `SplitText` to index them into solr

all the parsed records are added to mongo by bulk - processor: mongo_publisher

component: com.hurence.logisland.processor.datastore.BulkPut type: processor documentation: “indexes processed events in Mongo” configuration:

datastore.client.service: datastore_service

3. Launch the job

1. Run docker-compose

For this tutorial we will handle some apache logs with a splitText parser and send them to Elastiscearch. Launch your docker container with this command (we suppose you are in the root of the tar gz assembly) :

```
sudo docker-compose -f ./conf/docker-compose-index-apache-logs-es.yml up -d
```

Make sure all container are running and that there is no error.

```
sudo docker-compose ps
```

Those containers should be visible and running

‘‘‘ CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES 0d9e02b22c38 docker.elastic.co/kibana/kibana:5.4.0 “/bin/sh -c /usr/loc…” 13 seconds ago Up 8 seconds 0.0.0.0:5601->5601/tcp conf_kibana_1 ab15f4b5198c docker.elastic.co/elasticsearch/elasticsearch:5.4.0 “/bin/bash bin/es-do…” 13 seconds ago Up 7 seconds 0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp conf_elasticsearch_1 a697e45d2d1a hurence/logisland:1.1.2 “tail -f bin/logisla…” 13 seconds ago Up 9 seconds 0.0.0.0:4050->4050/tcp, 0.0.0.0:8082->8082/tcp, 0.0.0.0:9999->9999/tcp conf_logisland_1 db80cdf23b45 hurence/zookeeper “/bin/sh -c ‘/usr/sb…” 13 seconds ago Up 10 seconds 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, 7072/tcp conf_zookeeper_1 7aa7a87dd16b hurence/kafka:0.10.2.2-scala-2.11 “start-kafka.sh” 13 seconds ago Up 5 seconds 0.0.0.0:9092->9092/tcp conf_kafka_1

‘‘‘

```
sudo docker logs conf_kibana_1
sudo docker logs conf_elasticsearch_1
sudo docker logs conf_logisland_1
sudo docker logs conf_zookeeper_1
sudo docker logs conf_kafka_1
```

Should not return errors or any suspicious messages

2. Initializing mongo db

Note: You have to create the db logisland with the collection apache.

```
# open the mongo shell inside mongo container
sudo docker exec -ti conf_mongo_1 mongo

> use logisland
switched to db logisland

> db.apache.insert({src_ip:"19.123.12.67", identd:"-", user:"-", bytes_out:12344,␣
↪http_method:"POST", http_version:"2.0", http_query:"/logisland/is/so?great=true",
↪http_status:"404" })
WriteResult({ "nInserted" : 1 })

> db.apache.find()
```

{ “_id” : ObjectId(“5b4f3c4a5561b53b7e862b57”), “src_ip” : “19.123.12.67”, “identd” : “-”, “user” : “-”, “bytes_out” : 12344, “http_method” : “POST”, “http_version” : “2.0”, “http_query” : “/logisland/is/so?great=true”, “http_status” : “404” }

3. Run logisland job

you can now run the job inside the logisland container

```
sudo docker exec -ti conf_logisland_1 ./bin/logisland.sh --conf ./conf/index-apache-
↪logs-mongo.yml
```

The last logs should be something like :

2019-03-19 16:08:47 INFO StreamProcessingRunner:95 - awaitTermination for engine 1 2019-03-19 16:08:47 WARN SparkContext:66 - Using an existing SparkContext; some configuration may not take effect.

4. Inject some Apache logs into the system

Now we're going to send some logs to `logisland_raw` Kafka topic.

If you don't have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

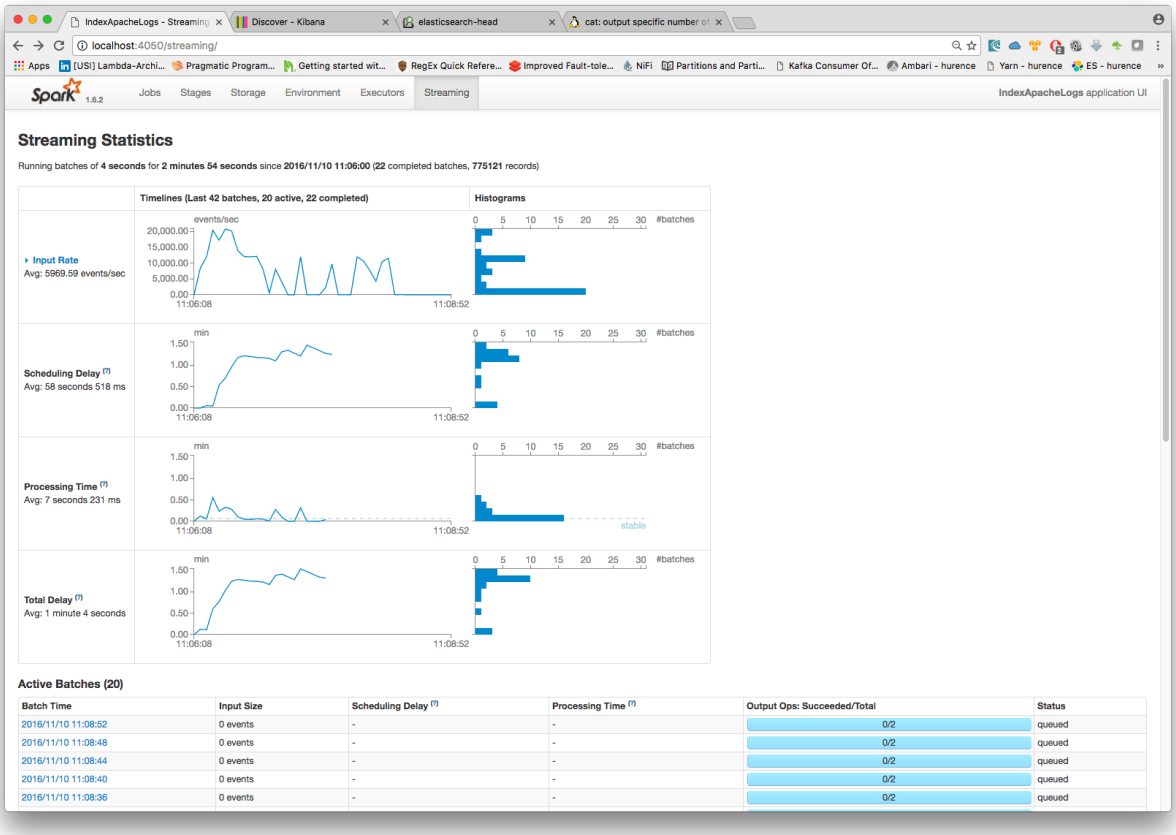
Let's send the first 500 lines of NASA http access over July 1995 to LogIsland with kafka scripts (available in our logisland container) to `logisland_raw` Kafka topic.

In another terminal run those commands

```
sudo docker exec -ti conf_logisland_1 bash
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -n 500 NASA_access_log_Jul95 | ${KAFKA_HOME}/bin/kafka-console-producer.sh --
↪broker-list kafka:9092 --topic logisland_raw
```

5. Monitor your spark jobs and Kafka topics

Now go to http://localhost:4050/streaming/ to see how fast Spark can process your data



6. Inspect the logs

With mongo you can directly use the shell:

```
> db.apache.find()
```

{ “_id” : “507adf3e-3162-4ff0-843a-253e01a6df69”, “src_ip” : “129.94.144.152”, “record_id” : “507adf3e-3162-4ff0-843a-253e01a6df69”, “http_method” : “GET”, “record_value” : “129.94.144.152 - - [01/Jul/1995:00:00:17 -0400] “GET /images/ksclogo-medium.gif HTTP/1.0” 304 0”, “http_query” : “/images/ksclogo-medium.gif”, “bytes_out” : “0”, “identd” : “-”, “http_version” : “HTTP/1.0”, “http_status” : “304”, “record_time” : NumberLong(“804571.1.4.1”), “user” : “-”, “record_type” : “apache_log” } { “_id” : “c44a9d09-52b9-4ada-8126-39c70c90fdd3”, “src_ip” : “ppp-mia-30.shadow.net”, “record_id” : “c44a9d09-52b9-4ada-8126-39c70c90fdd3”, “http_method” : “GET”, “record_value” : “ppp-mia-30.shadow.net - - [01/Jul/1995:00:00:27 -0400] “GET / HTTP/1.0” 200 7074”, “http_query” : “/”, “bytes_out” : “7074”, “identd” : “-”, “http_version” : “HTTP/1.0”, “http_status” : “200”, “record_time” : NumberLong(“804571.4.100”), “user” : “-”, “record_type” : “apache_log” } . . .

3. Stop the job

You can Ctr+c the console where you launched logisland job. Then to kill all containers used run :

```
sudo docker-compose -f ./conf/docker-compose-index-apache-logs-es.yml down
```

Make sure all container have disappeared.

```
sudo docker ps
```

1.2.6 Apache logs indexing with solr

In the following getting started tutorial we’ll drive you through the process of Apache log mining with LogIsland platform. The final data will be stored in solr

This tutorial is very similar to :

- Apache logs indexing into mongodb
- Apache logs indexing into elasticsearch

Note: Please note that you should not launch silmutaneously several docker-compose because we are exposing local port in them. So running several at the same time would be conflicting. So be sure to have killed all your currently running containers.

1.Install required components

- You either use docker-compose with available docker-compose-index-apache-logs-es.yml file in the tar.gz assembly in the conf folder.

  In this case you can skip this section

- Or you can launch the job in your cluster, but in this case you will have to make changes to job conf file so it works in your environment.

  In this case please make sure to already have installed solr modules (depending on which base you will use).

If not you can just do it through the components.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-service-mongodb-client:1.1.2
```

Note: In the following sections we will use docker-compose to run the job. (please install it before pursuing if you are not using your own cluster)

2. Logisland job setup

The logisland job that we will use is **./conf/index-apache-logs-solr.yml** The logisland docker-compose file that we will use is **./conf/docker-compose-index-apache-logs-solr.yml**

We will start by explaining each part of the config file.

An Engine is needed to handle the stream processing. This `conf/index-apache-logs-solr.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode with 2 cpu cores and 2G of RAM.

```
engine:
  component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
  type: engine
  documentation: Index some apache logs with logisland
  configuration:
    spark.app.name: IndexApacheLogsDemo
    spark.master: local[2]
    spark.driver.memory: 1G
    spark.driver.cores: 1
    spark.executor.memory: 2G
    spark.executor.instances: 4
    spark.executor.cores: 2
    spark.yarn.queue: default
    spark.yarn.maxAppAttempts: 4
    spark.yarn.am.attemptFailuresValidityInterval: 1h
    spark.yarn.max.executor.failures: 20
    spark.yarn.executor.failuresValidityInterval: 1h
    spark.task.maxFailures: 8
    spark.serializer: org.apache.spark.serializer.KryoSerializer
    spark.streaming.batchDuration: 1000
    spark.streaming.backpressure.enabled: false
    spark.streaming.unpersist: false
    spark.streaming.blockInterval: 500
    spark.streaming.kafka.maxRatePerPartition: 3000
    spark.streaming.timeout: -1
    spark.streaming.kafka.maxRetries: 3
    spark.streaming.ui.retainedBatches: 200
    spark.streaming.receiver.writeAheadLog.enable: false
    spark.ui.port: 4050
```

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job, here an Solr service that will be used later in the `TODO` processor.

```
# Datastore service using Solr 6.6.2 - 5.5.5 also available
- controllerService: datastore_service
  component: com.hurence.logisland.service.solr.Solr_6_6_2_ClientService
  type: service
```

(continues on next page)

(continued from previous page)

```
  documentation: "SolR 6.6.2 service"
  configuration:
    solr.cloud: false
    solr.connection.string: ${SOLR_CONNECTION}
    solr.collection: solr-apache-logs
    solr.concurrent.requests: 4
    flush.interval: 2000
    batch.size: 1000
```

Note: As you can see it uses environment variable so make sure to set them. (if you use the docker-compose file of this tutorial it is already done for you)

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our ouput records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that converts raw apache logs into structured log records
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: ${KAFKA_BROKERS}
    kafka.zookeeper.quorum: ${ZK_QUORUM}
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

Note: As you can see it uses environment variable so make sure to set them. (if you use the docker-compose file of this tutorial it is already done for you)

Within this stream a `SplitText` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
# parse apache logs into logisland records
- processor: apache_parser
  component: com.hurence.logisland.processor.SplitText
  type: parser
  documentation: a parser that produce events from an apache log REGEX
  configuration:
    record.type: apache_log
```

(continues on next page)

(continued from previous page)

```
    value.regex: (\S+)\s+(\S+)\s+(\S+)\s+\[([\w:\/]+\s[+\-]\d{4})\]\s+
↪"(\S+)\s+(\S+)\s*(\S*)"\s+(\S+)\s+(\S+)
    value.fields: src_ip,identd,user,record_time,http_method,http_query,http_version,
↪http_status,bytes_out
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

The second processor will handle `Records` produced by the `SplitText` to index them into solr

```
# all the parsed records are added to solr by bulk
- processor: solr_publisher
  component: com.hurence.logisland.processor.datastore.BulkPut
  type: processor
  documentation: "indexes processed events in SolR"
  configuration:
    datastore.client.service: datastore_service
```

3. Launch the job

1. Run docker-compose

For this tutorial we will handle some apache logs with a splitText parser and send them to Elastiscearch. Launch your docker container with this command (we suppose you are in the root of the tar gz assembly) :

```
sudo docker-compose -f ./conf/docker-compose-index-apache-logs-solr.yml up -d
```

Make sure all container are running and that there is no error.

```
sudo docker-compose ps
```

Those containers should be visible and running

‘‘‘ CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES 0d9e02b22c38 docker.elastic.co/kibana/kibana:5.4.0 “/bin/sh -c /usr/loc…” 13 seconds ago Up 8 seconds 0.0.0.0:5601->5601/tcp conf_kibana_1 ab15f4b5198c docker.elastic.co/elasticsearch/elasticsearch:5.4.0 “/bin/bash bin/es-do…” 13 seconds ago Up 7 seconds 0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp conf_elasticsearch_1 a697e45d2d1a hurence/logisland:1.1.2 “tail -f bin/logisla…” 13 seconds ago Up 9 seconds 0.0.0.0:4050->4050/tcp, 0.0.0.0:8082->8082/tcp, 0.0.0.0:9999->9999/tcp conf_logisland_1 db80cdf23b45 hurence/zookeeper “/bin/sh -c ‘/usr/sb…” 13 seconds ago Up 10 seconds 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, 7072/tcp conf_zookeeper_1 7aa7a87dd16b hurence/kafka:0.10.2.2-scala-2.11 “start-kafka.sh” 13 seconds ago Up 5 seconds 0.0.0.0:9092->9092/tcp conf_kafka_1

‘‘‘

```
sudo docker logs conf_kibana_1
sudo docker logs conf_elasticsearch_1
sudo docker logs conf_logisland_1
sudo docker logs conf_zookeeper_1
sudo docker logs conf_kafka_1
```

Should not return errors or any suspicious messages

2. Initializing solr db

We will now set up our solr database. First create the 'solr-apache-logs' collection

```
sudo docker exec -it --user=solr conf_solr_1 bin/solr create_core -c solr-apache-logs
```

The core/collection should have thos fields (corresponding to apache logs parsed fields) [src_ip, identd, user, bytes_out,] http_method, http_version, http_query, http_status

Otherwise for simplicity you can add a dynamic field called '*' and of type string for this collection with the web ui : http://localhost:8983/solr

Select the solr-apache-logs collection, go to schema and add your fields.

3. Run logisland job

you can now run the job inside the logisland container

```
sudo docker exec -ti conf_logisland_1 ./bin/logisland.sh --conf ./conf/index-apache-
↪logs-solr.yml
```

The last logs should be something like :

2019-03-19 16:08:47 INFO StreamProcessingRunner:95 - awaitTermination for engine 1 2019-03-19 16:08:47 WARN SparkContext:66 - Using an existing SparkContext; some configuration may not take effect.

4. Inject some Apache logs into the system

Now we're going to send some logs to logisland_raw Kafka topic.

If you don't have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

Let's send the first 500 lines of NASA http access over July 1995 to LogIsland with kafka scripts (available in our logisland container) to logisland_raw Kafka topic.

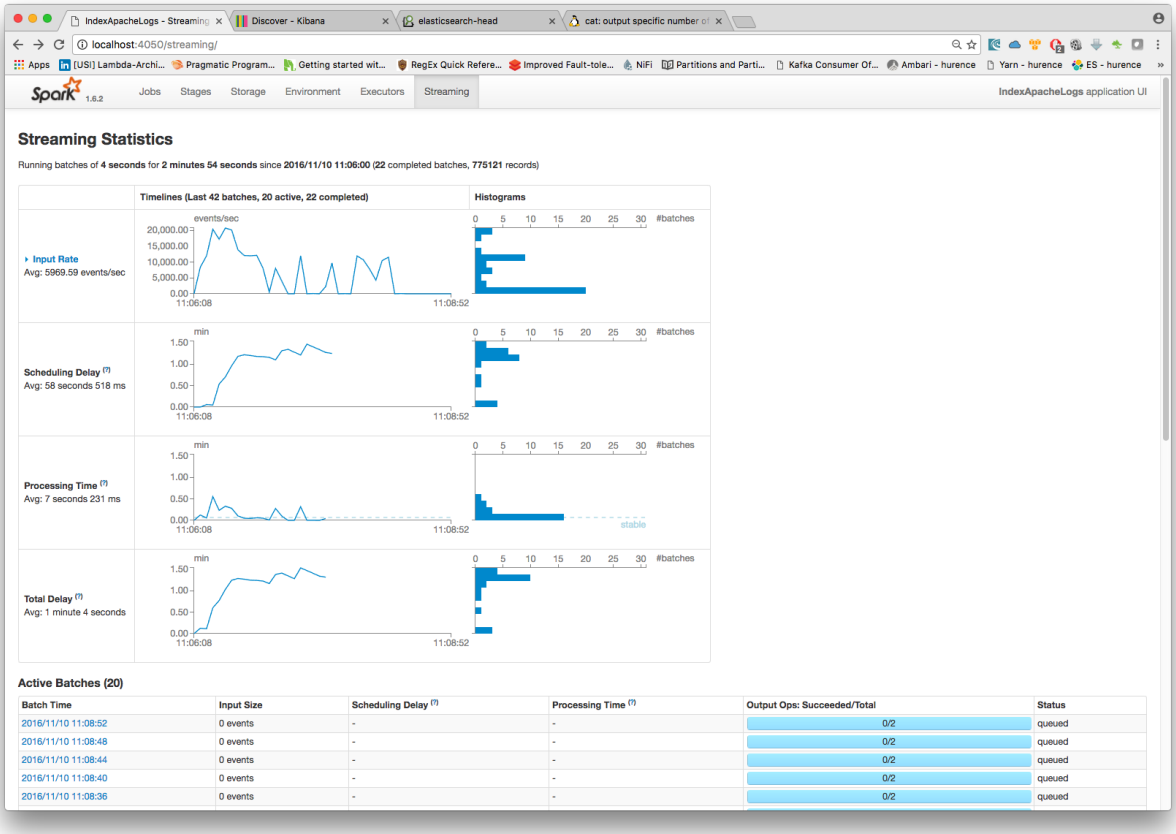
In another terminal run those commands

```
sudo docker exec -ti conf_logisland_1 bash
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -n 500 NASA_access_log_Jul95 | ${KAFKA_HOME}/bin/kafka-console-producer.sh --
↪broker-list kafka:9092 --topic logisland_raw
```

The logisland job should output logs, verify that there is no error, otherwise there is chances that your solr collection is not well configured.

5. Monitor your spark jobs and Kafka topics

Now go to http://localhost:4050/streaming/ to see how fast Spark can process your data

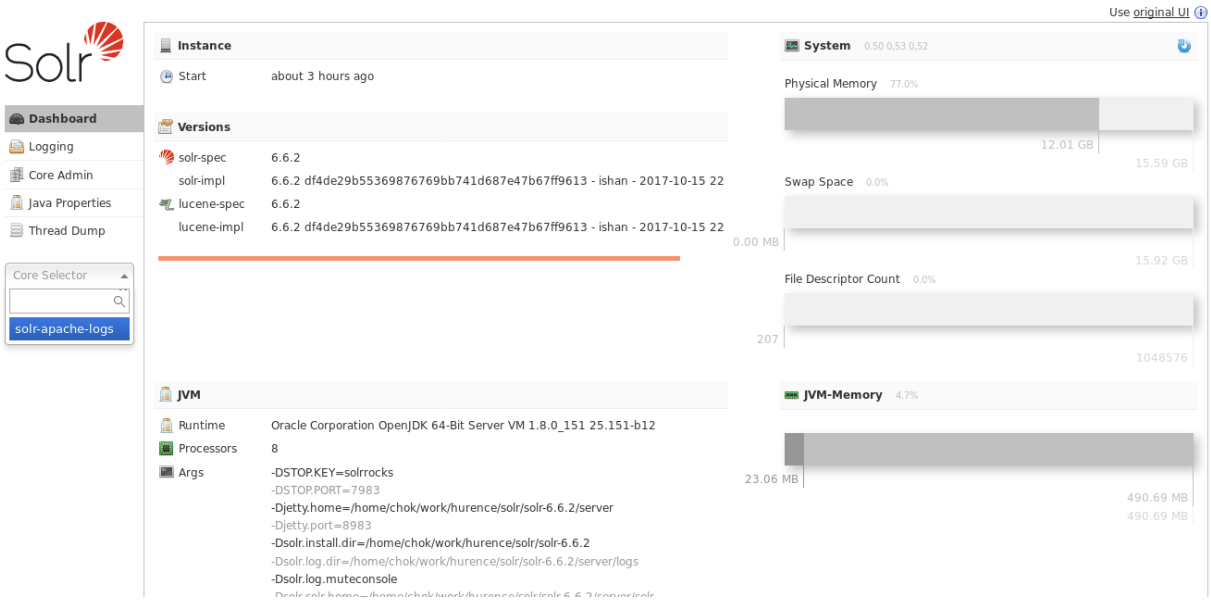


6. Inspect the logs

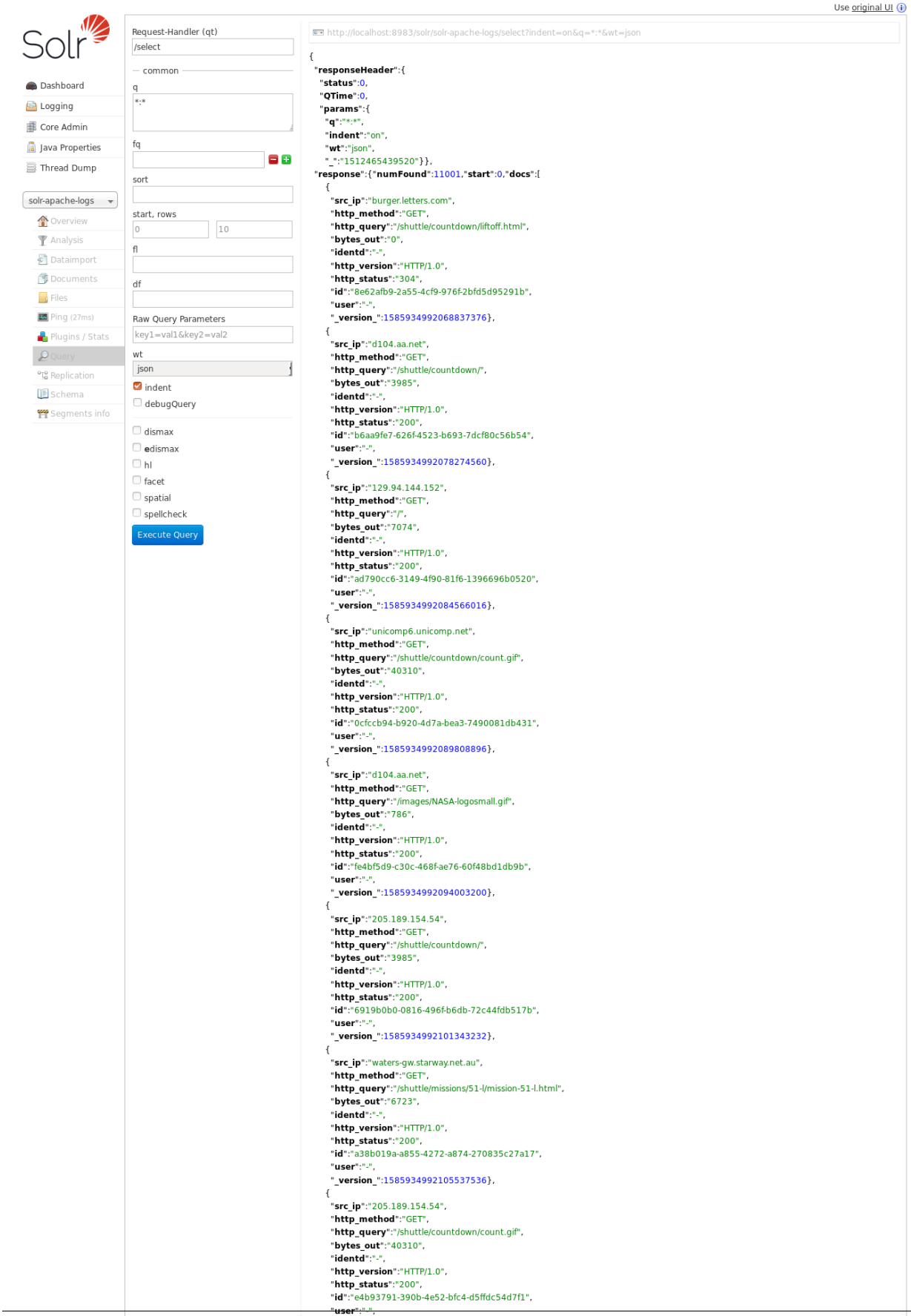
With Solr, you can directly use the solr web ui.

Open up your browser and go to http://localhost:8983/solr and you should be able to view your apache logs.

In non cloud mode, use the core selector, to select the core `solr-apache-logs` :



Then, go to query and by clicking to Execute Query, you will see some data from your Apache logs :


Use original UI
Solr
Dashboard
Logging
Core Admin
Java Properties
Thread Dump
solr-apache-logs
Overview
Analysis
Dataimport
Documents
Files
Ping (27ms)
Plugins / Stats
Query
Replication
Schema
Segments info
Request-Handler (qt)
/select
common
q
:
fq
sort
start, rows
0
10
fl
df
Raw Query Parameters
key1=val1&key2=val2
wt
json
indent
debugQuery
dismax
edismax
hl
facet
spatial
spellcheck
Execute Query
http://localhost:8983/solr/solr-apache-logs/select?indent=on&q=*:*&wt=json
{
"responseHeader":{
"status":0,
"QTime":0,
"params":{
"q":"*:*",
"indent":"on",
"wt":"json",
"_":"1512465439520"}},
"response":{"numFound":11001,"start":0,"docs":[
{
"src_ip":"burger.letters.com",
"http_method":"GET",
"http_query":"/shuttle/countdown/liftoff.html",
"bytes_out":"0",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"304",
"id":"8e62afb9-2a55-4cf9-976f-2bfd5d95291b",
"user":"-",
"_version_":1585934992068837376},
{
"src_ip":"d104.aa.net",
"http_method":"GET",
"http_query":"/shuttle/countdown/",
"bytes_out":"3985",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"200",
"id":"b6aa9fe7-626f-4523-b693-7dcf80c56b54",
"user":"-",
"_version_":1585934992078274560},
{
"src_ip":"129.94.144.152",
"http_method":"GET",
"http_query":"/",
"bytes_out":"7074",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"200",
"id":"ad790cc6-3149-4f90-81f6-1396696b0520",
"user":"-",
"_version_":1585934992084566016},
{
"src_ip":"unicomp6.unicomp.net",
"http_method":"GET",
"http_query":"/shuttle/countdown/count.gif",
"bytes_out":"40310",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"200",
"id":"0cfccb94-b920-4d7a-bea3-7490081db431",
"user":"-",
"_version_":1585934992089808896},
{
"src_ip":"d104.aa.net",
"http_method":"GET",
"http_query":"/images/NASA-logosmall.gif",
"bytes_out":"786",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"200",
"id":"fe4bf5d9-c30c-468f-ae76-60f48bd1db9b",
"user":"-",
"_version_":1585934992094003200},
{
"src_ip":"205.189.154.54",
"http_method":"GET",
"http_query":"/shuttle/countdown/",
"bytes_out":"3985",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"200",
"id":"6919b0b0-0816-496f-b6db-72c44fdb517b",
"user":"-",
"_version_":1585934992101343232},
{
"src_ip":"waters-gw.starway.net.au",
"http_method":"GET",
"http_query":"/shuttle/missions/51-l/mission-51-l.html",
"bytes_out":"6723",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"200",
"id":"a38b019a-a855-4272-a874-270835c27a17",
"user":"-",
"_version_":1585934992105537536},
{
"src_ip":"205.189.154.54",
"http_method":"GET",
"http_query":"/shuttle/countdown/count.gif",
"bytes_out":"40310",
"identd":"-",
"http_version":"HTTP/1.0",
"http_status":"200",
"id":"e4b93791-390b-4e52-bfc4-d5ffdc54d7f1",
"user":"-",
"_version_":1585934992110780416},
{
"src_ip":"unicomp6.unicomp.net",
"http_method":"GET",
"http_query":"/shuttle/countdown/",
"bytes_out":"3985",
"identd":"-",
"http_version":"HTTP/1.0",

3. Stop the job

You can Ctr+c the console where you launched logisland job. Then to kill all containers used run :

```
sudo docker-compose -f ./conf/docker-compose-index-apache-logs-solr.yml down
```

Make sure all container have disappeared.

```
sudo docker ps
```

1.2.7 Store Apache logs to Redis K/V store

In the following getting started tutorial we'll drive you through the process of Apache log mining with LogIsland platform.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

Note, it is possible to store data in different datastores. In this tutorial, we will see the case of Redis, if you need more in-depth explanations you can read the previous tutorial on indexing apache logs to elasticsearch or solr : **'index-apache-logs.html'_** .

1. Logisland job setup

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here :

```
docker exec -i -t logisland vim conf/store-to-redis.yml
```

We will start by explaining each part of the config file.

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job, here a Redis KV cache service that will be used later in the `BulkPut` processor.

```
- controllerService: datastore_service
  component: com.hurence.logisland.redis.service.RedisKeyValueCacheService
  type: service
  documentation: redis datastore service
  configuration:
    connection.string: localhost:6379
    redis.mode: standalone
    database.index: 0
    communication.timeout: 10 seconds
    pool.max.total: 8
    pool.max.idle: 8
    pool.min.idle: 0
    pool.block.when.exhausted: true
    pool.max.wait.time: 10 seconds
    pool.min.evictable.idle.time: 60 seconds
    pool.time.between.eviction.runs: 30 seconds
    pool.num.tests.per.eviction.run: -1
    pool.test.on.create: false
    pool.test.on.borrow: false
    pool.test.on.return: false
    pool.test.while.idle: true
    record.recordSerializer: com.hurence.logisland.serializer.JsonSerializer
```

Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our ouput records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that converts raw apache logs into structured log records
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

Within this stream a `SplitText` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
# parse apache logs
- processor: apache_parser
  component: com.hurence.logisland.processor.SplitText
  type: parser
  documentation: a parser that produce events from an apache log REGEX
  configuration:
    value.regex: (\S+)\s+(\S+)\s+(\S+)\s+\[([\w:\/]+\s[+\-]\d{4})\]\s+
↪"(\S+)\s+(\S+)\s*(\S*)"\s+(\S+)\s+(\S+)
    value.fields: src_ip,identd,user,record_time,http_method,http_query,http_version,
↪http_status,bytes_out
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

The second processor will handle `Records` produced by the `SplitText` to index them into datastore previously defined (Redis)

```
# all the parsed records are added to datastore by bulk
- processor: datastore_publisher
  component: com.hurence.logisland.processor.datastore.BulkPut
  type: processor
  documentation: "indexes processed events in datastore"
  configuration:
    datastore.client.service: datastore_service
```

2. Launch the script

For this tutorial we will handle some apache logs with a splitText parser and send them to Redis Connect a shell to your logisland container to launch the following streaming jobs.

For ElasticSearch :

```
docker exec -i -t logisland bin/logisland.sh --conf conf/store-to-redis.yml
```

3. Inject some Apache logs into the system

Now we're going to send some logs to `logisland_raw` Kafka topic.

We could setup a logstash or flume agent to load some apache logs into a kafka topic but there's a super useful tool in the Kafka ecosystem : kafkacat, a *generic command line non-JVM Apache Kafka producer and consumer* which can be easily installed.

If you don't have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

Let's send the first 500000 lines of NASA http access over July 1995 to LogIsland with kafkacat to `logisland_raw` Kafka topic

```
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -500000 NASA_access_log_Jul95 | kafkacat -b sandbox:9092 -t logisland_raw
```

4. Inspect the logs

For this part of the tutorial we will use redis-py a Python client for Redis. You can install it by following instructions given on redis-py.

To install redis-py, simply:

```
$ sudo pip install redis
```

Getting Started, check if you can connect with Redis

```
>>> import redis
>>> r = redis.StrictRedis(host='localhost', port=6379, db=0)
>>> r.set('foo', 'bar')
>>> r.get('foo')
```

Then we want to grab some logs that have been collected to Redis. We first find some keys with a pattern and get the json content of one

```
>>> r.keys('1234*')
```

['123493eb-93df-4e57-a1c1-4a8e844fa92c', '123457d5-8ccc-4f0f-b4ba-d70967aa48eb', '12345e06-6d72-4ce8-8254-a7cc4bab5e31']

```
>>> r.get('123493eb-93df-4e57-a1c1-4a8e844fa92c')
```

‘{n “id” : “123493eb-93df-4e57-a1c1-4a8e844fa92c”,n “type” : “apache_log”,n “creationDate” : 804574829000,n “fields” : {n “src_ip” : “204.191.209.4”,n “record_id” : “123493eb-93df-4e57-a1c1-4a8e844fa92c”,n “http_method” : “GET”,n “http_query” : “/images/WORLD-logosmall.gif”,n “bytes_out” : “669”,n “identd” : “-“,n “http_version” : “HTTP/1.0”,n “record_raw_value” : “204.191.209.4 - - [01/Jul/1995:01:00:29 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 200 669”,n “http_status” : “200”,n “record_time” : 804574829000,n “user” : “-“,n “record_type” : “apache_log”n }n}’

```
>>> import json
>>> record = json.loads(r.get('123493eb-93df-4e57-a1c1-4a8e844fa92c'))
>>> record['fields']['bytes_out']
```

1.2.8 Threshold based alerting on Apache logs with Redis K/V store

In a previous tutorial we saw how to use Redis K/V store as a cache storage. In this one we will practice the use of *ComputeTag*, *CheckThresholds* and *CheckAlerts* processor in conjunction with this Redis Cache.

The following job is made of 2 streaming parts :

1. A main stream which parses Apache logs and store them to a Redis cache .
2. A timer based stream which compute some new tags values based on cached records, check some thresholds cross and send alerts if needed.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

The full logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here :

```
docker exec -i -t conf_logisland_1 vim conf/threshold-alerting.yml
```

We will start by explaining each part of the config file.

1. Controller service part

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job, here a Redis KV cache service that will be used later in the `BulkPut` processor.

```
- controllerService: datastore_service
  component: com.hurence.logisland.redis.service.RedisKeyValueCacheService
  type: service
  documentation: redis datastore service
  configuration:
    connection.string: localhost:6379
    redis.mode: standalone
    database.index: 0
    communication.timeout: 10 seconds
    pool.max.total: 8
    pool.max.idle: 8
    pool.min.idle: 0
    pool.block.when.exhausted: true
    pool.max.wait.time: 10 seconds
    pool.min.evictable.idle.time: 60 seconds
```

(continues on next page)

(continued from previous page)

```
    pool.time.between.eviction.runs: 30 seconds
    pool.num.tests.per.eviction.run: -1
    pool.test.on.create: false
    pool.test.on.borrow: false
    pool.test.on.return: false
    pool.test.while.idle: true
    record.recordSerializer: com.hurence.logisland.serializer.JsonSerializer
```

2. First stream : parse logs and compute tags

Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic as Json serialized records.

```
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that converts raw apache logs into structured log records
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

Within this stream a `SplitText` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
- processor: apache_parser
  component: com.hurence.logisland.processor.SplitText
  type: parser
  documentation: a parser that produce events from an apache log REGEX
  configuration:
    value.regex: (\S+)\s+(\S+)\s+(\S+)\s+\[([\w:\/]+\s[+\-]\d{4})\]\s+
↪"(\S+)\s+(\S+)\s*(\S*)"\s+(\S+)\s+(\S+)
    value.fields: src_ip,identd,user,record_time,http_method,http_query,http_version,
↪http_status,bytes_out
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

the next processing step is to assign *bytes_out* field as *record_value*

```
- processor: normalize_fields
  component: com.hurence.logisland.processor.NormalizeFields
  type: parser
  documentation: change field name 'bytes_out' to `record_value`
  configuration:
    conflict.resolution.policy: overwrite_existing
    record_value: bytes_out
```

the we modify *record_id* to set its value as *src_ip* field.

```
- processor: modify_id
  component: com.hurence.logisland.processor.ModifyId
  type: parser
  documentation: change current id to src_ip
  configuration:
    id.generation.strategy: fromFields
    fields.to.hash: src_ip
    java.formatter.string: "%1$s"
```

now we'll remove all the unwanted fields

```
- processor: remove_fields
  component: com.hurence.logisland.processor.RemoveFields
  type: parser
  documentation: remove useless fields
  configuration:
    fields.to.remove: src_ip,identd,user,http_method,http_query,http_version,http_
↪status,bytes_out
```

and then cast *record_value* as a double

```
- processor: cast
  component: com.hurence.logisland.processor.ConvertFieldsType
  type: parser
  documentation: cast values
  configuration:
    record_value: double
```

The next processing step wil compute a dynamic Tag value from a Javascript expression. Here a new record with an *record_id* set to *computed1* and as a *record_value* the resulting expression of *cache("logisland.hurence.com").value * 10.2*

```
- processor: compute_tag
  component: com.hurence.logisland.processor.alerting.ComputeTags
  type: processor
  documentation: |
    compute tags from given formulas.
    each dynamic property will return a new record according to the formula definition
    the record name will be set to the property name
    the record time will be set to the current timestamp
  configuration:
    datastore.client.service: datastore_service
    output.record.type: computed_tag
    max.cpu.time: 500
    max.memory: 64800000
    max.prepared.statements: 5
    allow.no.brace: false
    computed1: return cache("logisland.hurence.com").value * 10.2;
```

The last processor will handle all the `Records` of this stream to index them into datastore previously defined (Redis)

```
# all the parsed records are added to datastore by bulk
- processor: datastore_publisher
  component: com.hurence.logisland.processor.datastore.BulkPut
  type: processor
  documentation: "indexes processed events in datastore"
```

(continues on next page)

(continued from previous page)

```
  configuration:
    datastore.client.service: datastore_service
```

3. Second stream : check threshold cross and alerting

The second stream will read all the logs sent in `logisland_events` topic and push the processed outputs (threshold_cross & alerts records) into `logisland_alerts` topic as Json serialized records.

We won't comment the stream definition as it is really straightforward.

The first processor of this stream pipeline makes use of *CheckThresholds* component which will add a new record of type *threshold_cross* with a *record_id* set to *threshold1* if the JS expression *cache("computed1").value > 2000.0* is evaluated to true.

```
- processor: compute_thresholds
  component: com.hurence.logisland.processor.alerting.CheckThresholds
  type: processor
  documentation: |
    compute threshold cross from given formulas.
    each dynamic property will return a new record according to the formula definition
    the record name will be set to the property name
    the record time will be set to the current timestamp

    a threshold_cross has the following properties : count, time, duration, value
  configuration:
    datastore.client.service: datastore_service
    output.record.type: threshold_cross
    max.cpu.time: 100
    max.memory: 12800000
    max.prepared.statements: 5
    record.ttl: 300000
    threshold1: cache("computed1").value > 2000.0
```

```
- processor: compute_alerts1
  component: com.hurence.logisland.processor.alerting.CheckAlerts
  type: processor
  documentation: |
    compute threshold cross from given formulas.
    each dynamic property will return a new record according to the formula definition
    the record name will be set to the property name
    the record time will be set to the current timestamp
  configuration:
    datastore.client.service: datastore_service
    output.record.type: medium_alert
    alert.criticity: 1
    max.cpu.time: 100
    max.memory: 12800000
    max.prepared.statements: 5
    profile.activation.condition: cache("threshold1").value > 3000.0
    alert1: cache("threshold1").duration > 50.0
```

The last processor will handle all the `Records` of this stream to index them into datastore previously defined (Redis)

```
- processor: datastore_publisher
  component: com.hurence.logisland.processor.datastore.BulkPut
```

(continues on next page)

(continued from previous page)

```
  type: processor
  documentation: "indexes processed events in datastore"
  configuration:
    datastore.client.service: datastore_service
```

4. Launch the script

Connect a shell to your logisland container to launch the following streaming jobs.

```
docker exec -i -t conf_logisland_1 bin/logisland.sh --conf conf/threshold-alerting.yml
```

5. Inject some Apache logs into the system

Now we're going to send some logs to `logisland_raw` Kafka topic.

We could setup a logstash or flume agent to load some apache logs into a kafka topic but there's a super useful tool in the Kafka ecosystem : kafkacat, a *generic command line non-JVM Apache Kafka producer and consumer* which can be easily installed.

If you don't have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

Let's send the first 500000 lines of NASA http access over July 1995 to LogIsland with kafkacat to `logisland_raw` Kafka topic

```
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -500000 NASA_access_log_Jul95 | kafkacat -b sandbox:9092 -t logisland_raw
```

6. Inspect the logs and alerts

For this part of the tutorial we will use redis-py a Python client for Redis. You can install it by following instructions given on redis-py.

To install redis-py, simply:

```
$ sudo pip install redis
```

Getting Started, check if you can connect with Redis

```
>>> import redis
>>> r = redis.StrictRedis(host='localhost', port=6379, db=0)
>>> r.set('foo', 'bar')
>>> r.get('foo')
```

Then we want to grab some logs that have been collected to Redis. We first find some keys with a pattern and get the json content of one

```
>>> r.keys('1234*')
```

['123493eb-93df-4e57-a1c1-4a8e844fa92c', '123457d5-8ccc-4f0f-b4ba-d70967aa48eb', '12345e06-6d72-4ce8-8254-a7cc4bab5e31']

```
>>> r.get('123493eb-93df-4e57-a1c1-4a8e844fa92c')
```

'{n "id" : "123493eb-93df-4e57-a1c1-4a8e844fa92c",n "type" : "apache_log",n "creationDate" : 804574829000,n "fields" : {n "src_ip" : "204.191.209.4",n "record_id" : "123493eb-93df-4e57-a1c1-4a8e844fa92c",n "http_method" : "GET",n "http_query" : "/images/WORLD-logosmall.gif",n "bytes_out" : "669",n "identd" : "-",n "http_version" : "HTTP/1.0",n "record_raw_value" : "204.191.209.4 - - [01/Jul/1995:01:00:29 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 200 669",n "http_status" : "200",n "record_time" : 804574829000,n "user" : "-",n "record_type" : "apache_log"n }n}'

```
>>> import json
>>> record = json.loads(r.get('123493eb-93df-4e57-a1c1-4a8e844fa92c'))
>>> record['fields']['bytes_out']
```

1.2.9 Alerting & Query Matching

In the following tutorial we'll learn how to raise custom alerts on some http traffic (apache log records) based on lucene matching query criterion.

We assume that you already know how to parse and ingest Apache logs into logisland. If it's not the case please refer to the previous Apache logs indexing tutorial. We will use mainly the MatchQuery Processor.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

1.Install required components

For this tutorial please make sure to already have installed elasticsearch modules.

If not you can just do it through the components.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_5_4_0-
↪client:1.1.2
```

2. Logisland job setup

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here :

```
docker exec -i -t logisland vim conf/match-queries.yml
```

We will start by explaining each part of the config file.

The stream contains two processors quite identical (the first one converts raw logs to records and the second one index records to ES) to those encountered in the previous Apache logs indexing tutorial tutorial .

The third one makes use of the MatchQuery Processor. This processor provides user with dynamic query registration. This queries are expressed in the Lucene syntax.

Note: Please read the Lucene syntax guide for supported operations.

This processor will tag the record with `blacklisted_host` field if the query `src_ip:(+alyssa +prodigy)` matches and tag `montana_host` if `src_ip:montana`

```
- processor: match_query
  component: com.hurence.logisland.processor.MatchQuery
  type: processor
  documentation: a parser that matches lucene queries on records
  configuration:
    policy.onmiss: forward
    policy.onmatch: all
    blacklisted_host: src_ip:(+alyssa +prodigy)
    montana_host: src_ip:montana
```

here is an example of matching record :

```
{
    "@timestamp": "1995-07-01T09:02:18+02:00",
    "alert_match_name": [
      "montana_host"
    ],
    "alert_match_query": [
      "src_ip:montana"
    ],
    "bytes_out": "8677",
    "http_method": "GET",
    "http_query": "/shuttle/missions/missions.html",
    "http_status": "200",
    "http_version": "HTTP/1.0",
    "identd": "-",
    "record_id": "8e861956-af54-49fd-9043-94c143fc5a19",
    "record_raw_value": "ril.usda.montana.edu - - [01/Jul/1995:03:02:18 -0400] \"GET /
↪shuttle/missions/missions.html HTTP/1.0\" 200 8677",
    "record_time": 804582138000,
    "record_type": "apache_log",
    "src_ip": "ril.usda.montana.edu",
    "user": "-"
  }
```

3. Launch the script

For this tutorial we will handle some apache logs with a splitText parser and send them to Elastiscearch Connect a shell to your logisland container to launch the following streaming jobs.

```
docker exec -i -t logisland bin/logisland.sh --conf conf/match-queries.yml
```

4. Inject some Apache logs into the system

Now we're going to send some logs to `logisland_raw` Kafka topic.

We could setup a logstash or flume agent to load some apache logs into a kafka topic but there's a super useful tool in the Kafka ecosystem : kafkacat, a *generic command line non-JVM Apache Kafka producer and consumer* which can be easily installed.

If you don't have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

Let's send the first 500000 lines of NASA http access over July 1995 to LogIsland with kafkacat to `logisland_raw` Kafka topic

```
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -500000 NASA_access_log_Jul95 | kafkacat -b sandbox:9092 -t logisland_raw
```

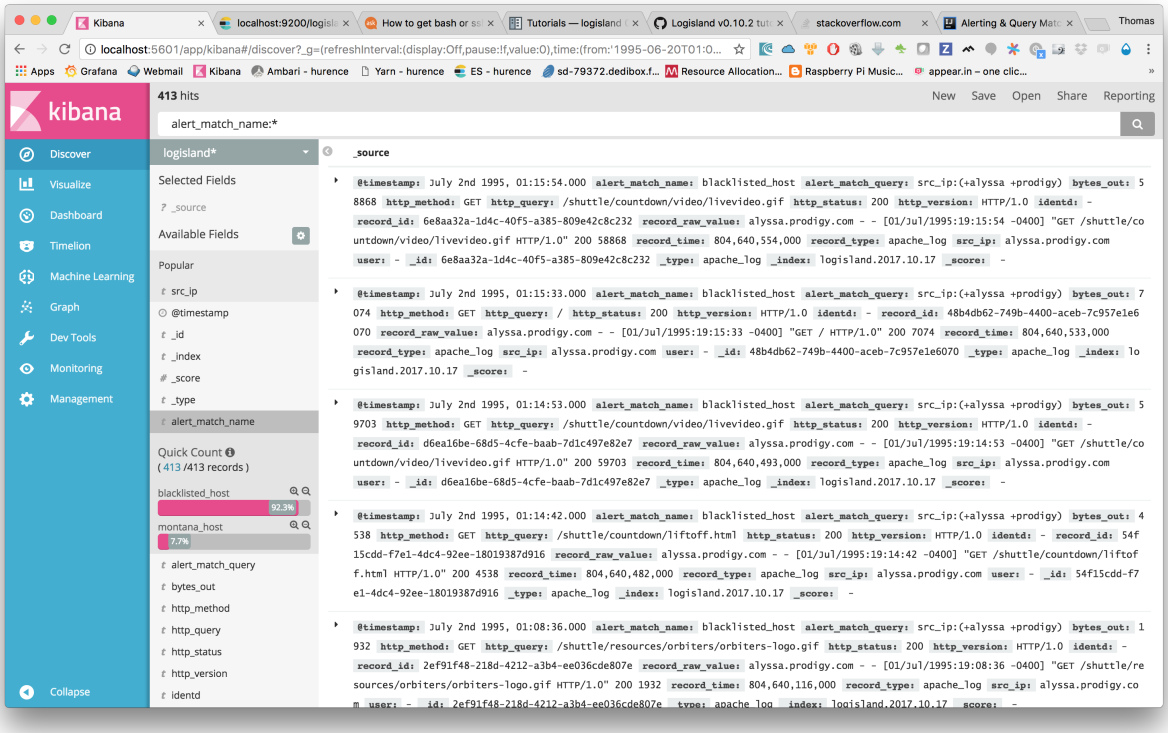
5. Check your alerts with Kibana

Check that you've match some criterias :

```
curl -XGET http://localhost:9200/logisland.2017.10.17/_search?pretty&q=alert_match_
↪name:montana_host
curl -XGET http://localhost:9200/logisland.2017.10.17/_search?pretty&q=alert_match_
↪name:blacklisted_host
```

Open up your browser and go to http://sandbox:5601/ and you should be able to explore your apache logs.

by adding filter on `alert_match_name:blacklisted_host` you'll only get request from `alyssa.prodigy.com` which is a host we where monitoring.



1.2.10 Event aggregation

In the following tutorial we'll learn how to generate time window metrics on some http traffic (apache log records) and how to raise custom alerts based on lucene matching query criterion.

We assume that you already know how to parse and ingest Apache logs into logisland. If it's not the case please refer to the previous Apache logs indexing tutorial. We will first add an SQLAggregator Stream to compute some metrics and then add a MatchQuery Processor.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

---

1.Install required components

For this tutorial please make sure to already have installed elasticsearch modules. If not you can just do it through the componentes.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_5_4_0-
↪client:1.1.2
```

2. Logisland job setup

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here :

```
docker exec -i -t logisland vim conf/aggregate-events.yml
```

We will start by explaining each part of the config file.

Our application will be composed of 4 streams :

The first one converts apache logs to typed records (please note the use of `ConvertFieldsType` processor)

The second one is the sql stream is a special one one use a KafkaRecordStreamSQLAggregator. This stream defines input/output topics names as well as Serializers, avro schema.

Note: The Avro schema is set for the input topic and must be same as the avro schema of the output topic for the stream that produces the records (please refer to Index Apache logs tutorial

---

The most important part of the *KafkaRecordStreamSQLAggregator* is its *sql.query* property which defines a query to apply on the incoming records for the given time window.

The following SQL query will be applied on sliding window of 10" of records.

```
SELECT count(*) AS connections_count, avg(bytes_out) AS avg_bytes_out, src_ip,␣
↪first(record_time) as record_time
FROM logisland_events
GROUP BY src_ip
ORDER BY connections_count DESC
LIMIT 20
```

which will consider `logisland_events` topic as SQL table and create 20 output Record with the fields avg_bytes_out, src_ip & record_time. the statement with record_time will ensure that the created Records will correspond to the effective input event time (not the actual time).

```
- stream: metrics_by_host
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamSQLAggregator
  type: stream
  documentation: a processor that links
  configuration:
    kafka.input.topics: logisland_events
    kafka.output.topics: logisland_aggregations
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 1
    window.duration: 10
    avro.input.schema: >
      {  "version":1,
         "type": "record",
         "name": "com.hurence.logisland.record.apache_log",
         "fields": [
           { "name": "record_errors",   "type": [ {"type": "array", "items": "string"}
↪,"null"] },
           { "name": "record_raw_key", "type": ["string","null"] },
           { "name": "record_raw_value", "type": ["string","null"] },
           { "name": "record_id",    "type": ["string"] },
           { "name": "record_time", "type": ["long"] },
           { "name": "record_type", "type": ["string"] },
           { "name": "src_ip",      "type": ["string","null"] },
           { "name": "http_method", "type": ["string","null"] },
           { "name": "bytes_out",   "type": ["long","null"] },
           { "name": "http_query",  "type": ["string","null"] },
           { "name": "http_version","type": ["string","null"] },
           { "name": "http_status", "type": ["string","null"] },
           { "name": "identd",      "type": ["string","null"] },
           { "name": "user",        "type": ["string","null"] }    ]}
    sql.query: >
      SELECT count(*) AS connections_count, avg(bytes_out) AS avg_bytes_out, src_ip
      FROM logisland_events
      GROUP BY src_ip
      ORDER BY event_count DESC
      LIMIT 20
    max.results.count: 1000
    output.record.type: top_client_metrics
```

Here we will compute every x seconds, the top twenty *src_ip* for connections count. The result of the query will be pushed into to *logisland_aggregations* topic as new *top_client_metrics* Record containing *connections_count* and *avg_bytes_out* fields.

the third match some criteria to send some alerts

```
- processor: match_query
  component: com.hurence.logisland.processor.MatchQuery
  type: processor
  documentation: a parser that produce alerts from lucene queries
  configuration:
```

(continues on next page)

(continued from previous page)

```
    numeric.fields: bytes_out,connections_count
    too_much_bandwidth: avg_bytes_out:[25000 TO 5000000]
    too_many_connections: connections_count:[150 TO 300]
    output.record.type: threshold_alert
```

3. Launch the script

For this tutorial we will handle some apache logs with a splitText parser and send them to Elastiscearch Connect a shell to your logisland container to launch the following streaming jobs.

```
docker exec -i -t logisland bin/logisland.sh --conf conf/aggregate-events.yml
```

4. Inject some Apache logs into the system

Now we're going to send some logs to `logisland_raw` Kafka topic.

We could setup a logstash or flume agent to load some apache logs into a kafka topic but there's a super useful tool in the Kafka ecosystem : kafkacat, a *generic command line non-JVM Apache Kafka producer and consumer* which can be easily installed.

If you don't have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

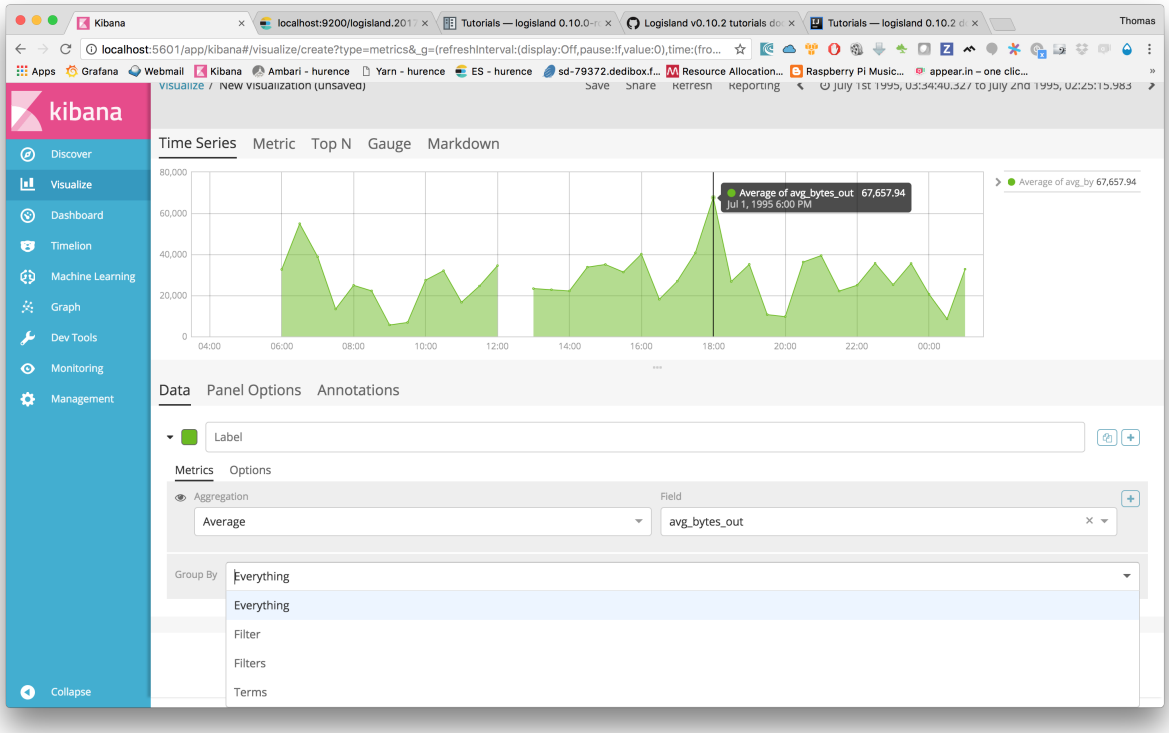
- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

Let's send the first 500000 lines of NASA http access over July 1995 to LogIsland with kafkacat to `logisland_raw` Kafka topic

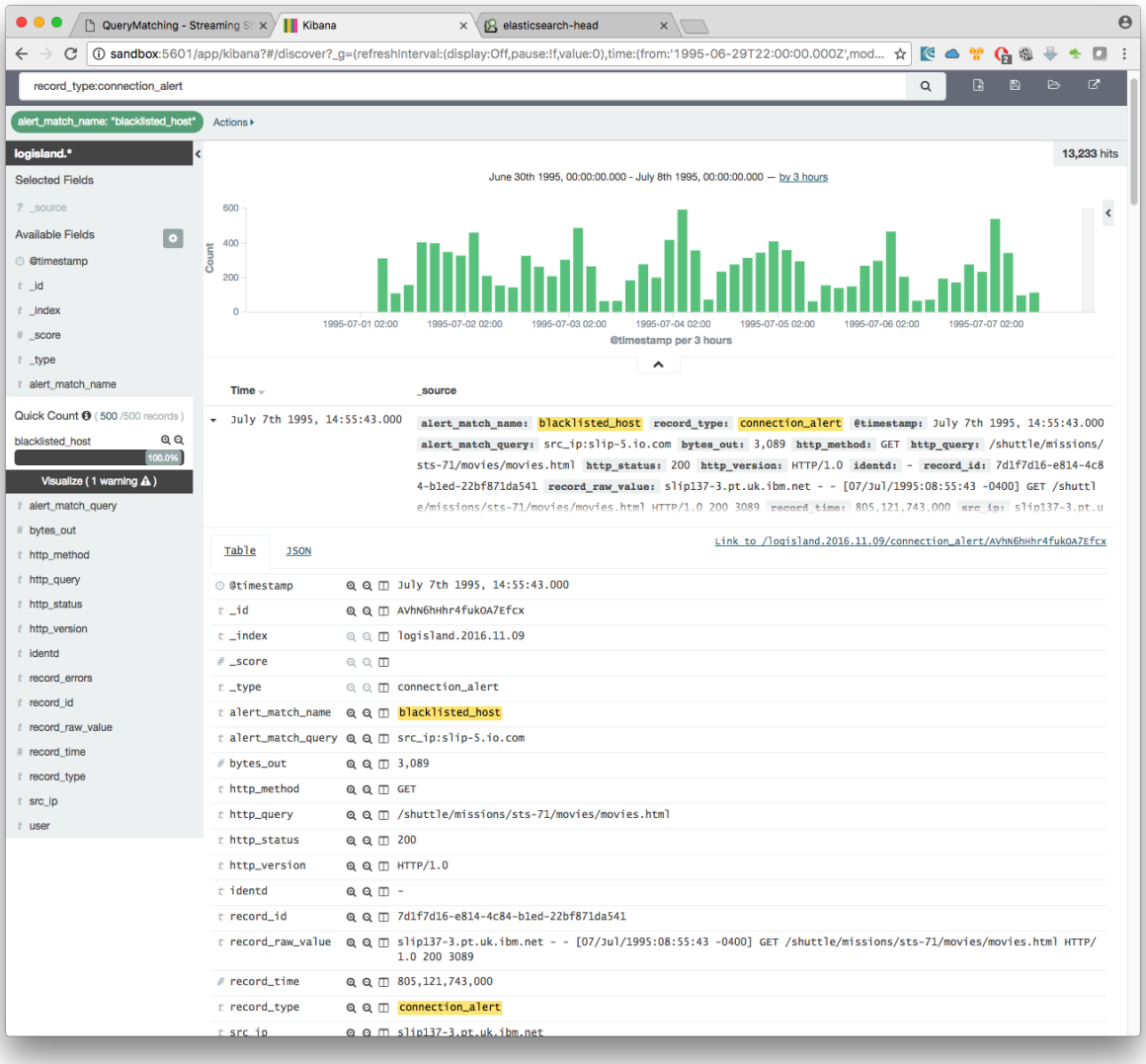
```
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -500000 NASA_access_log_Jul95 | kafkacat -b sandbox:9092 -t logisland_raw
```

5. Check your alerts with Kibana

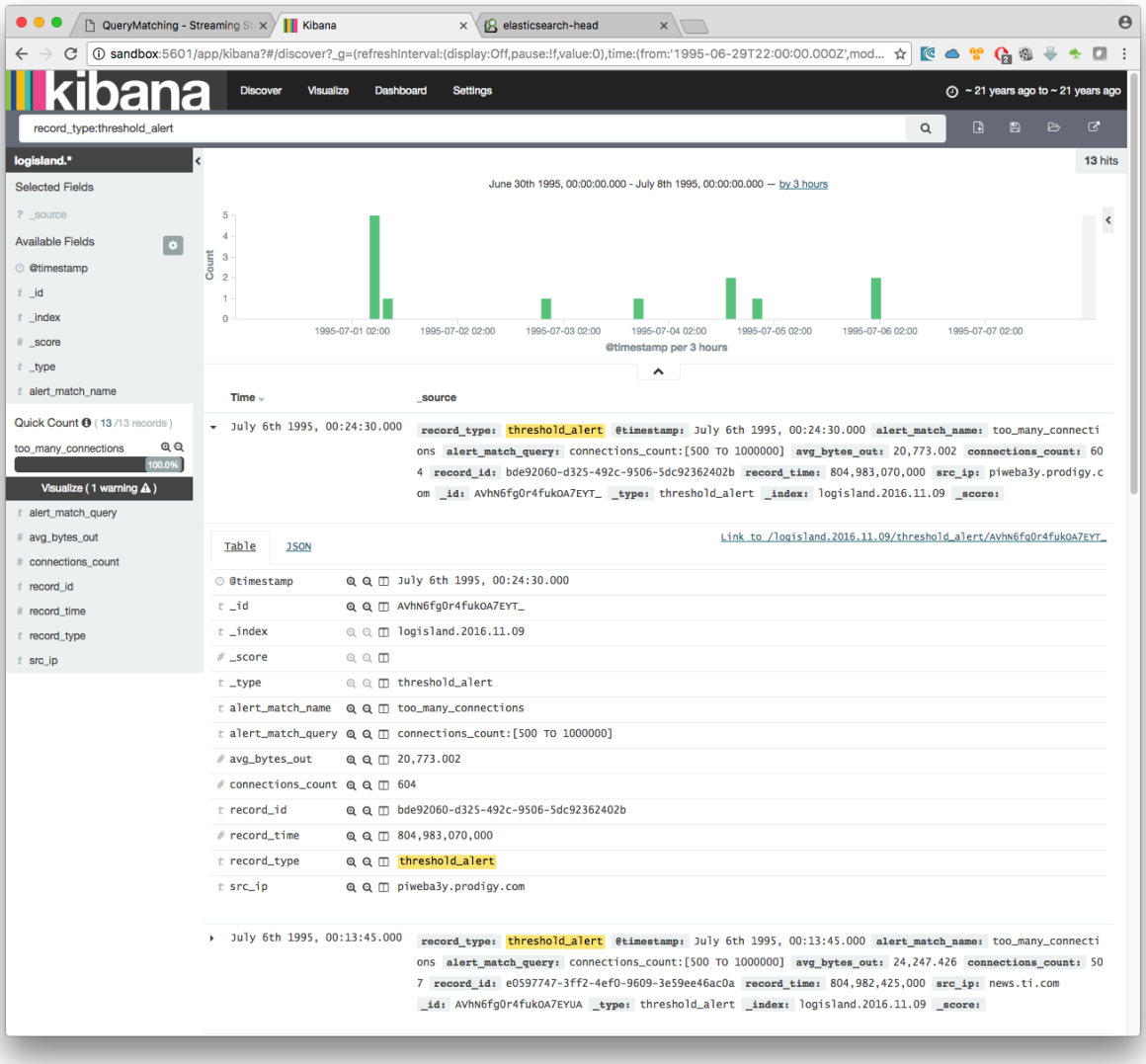
As we explore data logs from july 1995 we'll have to select an absolute time filter from 1995-06-30 to 1995-07-08 to see the events.



you can filter your events with `record_type:connection_alert` to get 71733 connections alerts matching your query



if we filter now on threshold alerts whith `record_type:threshold_alert` you'll get the 13 src_ip that have been catched by the threshold query.



1.2.11 Index Apache logs Enrichment

In the following tutorial we'll drive you through the process of enriching Apache logs with LogIsland platform.

One of the first steps when treating web access logs is to extract information from the User-Agent header string, in order to be able to classify traffic. The User-Agent string is part of the access logs from the web server (this is the last field in the example below).

Another step is to find the FQDN (full qualified domain name) from an ip address.

That string is packed with information from the visitor, when you know how to interpret it. However, the User-Agent string is not based on any standard, and it is not trivial to extract meaningful information from it. LogIsland provides a processor, based on the YAUAA library, that simplifies that treatement.

LogIsland provides a processor, based on InetAdress class from JDK 8, that use reverse Dns to determine FQDN from an IP.

Note: This class find FQDN from ip using IN-ADDR.ARPA (or IP6.ARPA for ipv6). If it finds a domain name, it

verifies that it matches back the same address ip in order to prevent against IP spoofing attack. If you want to return the ip anyway, you should implement a new plugin using another library as dnsjava for example or open an issue for asking this feature.

We will reuse the Docker container hosting all the LogIsland services from the previous tutorial, and add the User-Agent as well as the IpToFqdn processor to the stream

Note: You can download the latest release of logisland and the YAML configuration file for this tutorial which can be also found under *$LOGISLAND_HOME/conf* directory.

1. Start LogIsland as a Docker container

LogIsland is packaged as a Docker container that you can build yourself or pull from Docker Hub.

You can find the steps to start the Docker image and start the LogIsland server in the previous tutorial. However, you'll start the server with a different configuration file (that already includes the necessary modifications)

Install required components

For this tutorial please make sure to already have installed required modules.

If not you can just do it through the components.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_2_4_0-
↪client:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-processor-enrichment:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-processor-useragent:1.1.2
```

Stream 1 : modify the stream to analyze the User-Agent string

Note: You can either apply the modifications from this section to the file *conf/index-apache-logs.yml* ot directly use the file *conf/enrich-apache-logs.yml* that already includes them.

The stream needs to be modified to

```
* modify the regex to add the referer and the User-Agent strings for the SplitText␣
↪processor
* modify the Avro schema to include the new fields returned by the UserAgentProcessor
* include the processing of the User-Agent string after the parsing of the logs
* include the processor IpToFqdn after the ParserUserAgent
* include a cache service to use with IpToFqdn processor
```

The example below shows how to include all of the fields supported by the processor.

Note: It is possible to remove unwanted fields from both the processor configuration and the Avro schema

Once the configuration file is updated, LogIsland must be restarted with that new configuration file.

```
bin/logisland.sh --conf <new_configuration_file>
```

2. Inject some Apache logs into the system

Now we're going to send some logs to `logisland_raw` Kafka topic.

We could setup a logstash or flume agent to load some apache logs into a kafka topic but there's a super useful tool in the Kafka ecosystem : kafkacat, a *generic command line non-JVM Apache Kafka producer and consumer* which can be easily installed (and is already present in the docker image).

If you don't have your own httpd logs available, you can use some freely available log files from Elastic web site

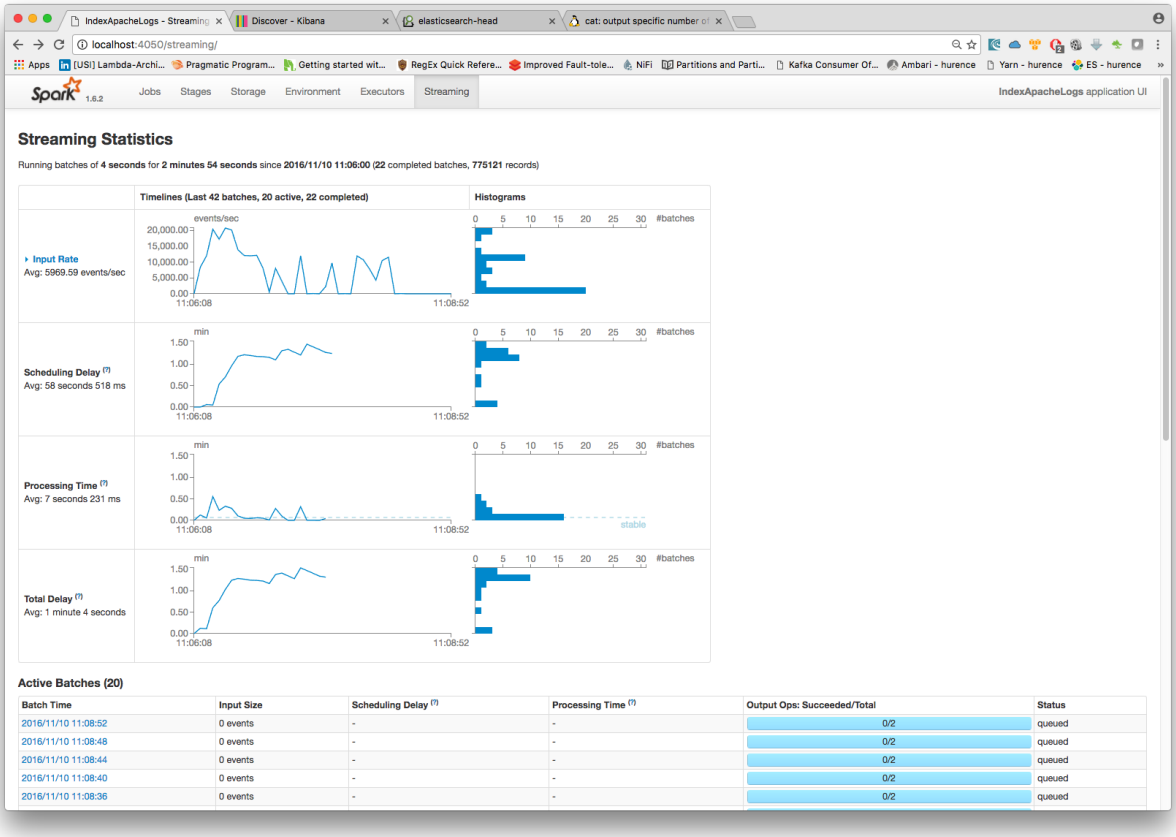
Let's send the first 500000 lines of access log to LogIsland with kafkacat to `logisland_raw` Kafka topic

```
docker exec -ti logisland bash
cd /tmp
wget https://raw.githubusercontent.com/elastic/examples/master/ElasticStack_apache/
↪apache_logs
head -500000 apache_logs | kafkacat -b sandbox:9092 -t logisland_raw
```

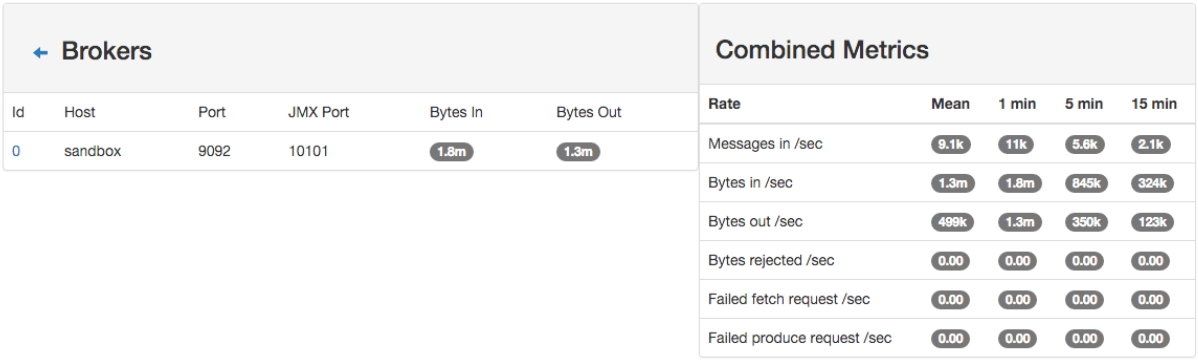
Note: The process should last around 280 seconds because reverse dns is a costly operation. After all data are processed, you can inject the same logs again and it should be very fast to process thanks to the cache that saved all matched ip.

3. Monitor your spark jobs and Kafka topics

Now go to http://sandbox:4050/streaming/ to see how fast Spark can process your data



Another tool can help you to tweak and monitor your processing http://sandbox:9000/



4. Use Kibana to inspect the logs

You've completed the enrichment of your logs using the User-Agent processor. The logs are now loaded into elasticSearch, in the following form :

```
curl -XGET http://localhost:9200/logisland.*/_search?pretty
```

```
{

    "_index": "logisland.2017.03.21",
    "_type": "apache_log",
    "_id": "4ca6a8b5-1a60-421e-9ae9-6c30330e497e",
    "_score": 1.0,
    "_source": {
        "@timestamp": "2015-05-17T10:05:43Z",
        "agentbuild": "Unknown",
        "agentclass": "Browser",
        "agentinformationemail": "Unknown",
        "agentinformationurl": "Unknown",
        "agentlanguage": "Unknown",
        "agentlanguagecode": "Unknown",
        "agentname": "Chrome",
        "agentnameversion": "Chrome 32.0.1700.77",
        "agentnameversionmajor": "Chrome 32",
        "agentsecurity": "Unknown",
        "agentuuid": "Unknown",
        "agentversion": "32.0.1700.77",
        "agentversionmajor": "32",
        "anonymized": "Unknown",
        "devicebrand": "Apple",
        "deviceclass": "Desktop",
        "devicecpu": "Intel",
        "devicefirmwareversion": "Unknown",
        "devicename": "Apple Macintosh",
        "deviceversion": "Unknown",
        "facebookcarrier": "Unknown",
        "facebookdeviceclass": "Unknown",
        "facebookdevicename": "Unknown",
        "facebookdeviceversion": "Unknown",
        "facebookfbop": "Unknown",
        "facebookfbss": "Unknown",
        "facebookoperatingsystemname": "Unknown",
        "facebookoperatingsystemversion": "Unknown",
        "gsainstallationid": "Unknown",
        "hackerattackvector": "Unknown",
        "hackertoolkit": "Unknown",
        "iecompatibilitynameversion": "Unknown",
        "iecompatibilitynameversionmajor": "Unknown",
        "iecompatibilityversion": "Unknown",
        "iecompatibilityversionmajor": "Unknown",
        "koboaffiliate": "Unknown",
        "koboplatformid": "Unknown",
        "layoutenginebuild": "Unknown",
        "layoutengineclass": "Browser",
        "layoutenginename": "Blink",
        "layoutenginenameversion": "Blink 32.0",
        "layoutenginenameversionmajor": "Blink 32",
        "layoutengineversion": "32.0",
        "layoutengineversionmajor": "32",
        "operatingsystemclass": "Desktop",
        "operatingsystemname": "Mac OS X",
        "operatingsystemnameversion": "Mac OS X 10.9.1",
        "operatingsystemversion": "10.9.1",
        "operatingsystemversionbuild": "Unknown",
```


(continues on next page)

(continued from previous page)

```
        "webviewappname": "Unknown",
        "webviewappnameversionmajor": "Unknown",
        "webviewappversion": "Unknown",
        "webviewappversionmajor": "Unknown",
        "bytes_out": 171717,
        "http_method": "GET",
        "http_query": "/presentations/logstash-monitorama-2013/images/kibana-
↪dashboard3.png",
        "http_referer": "http://semicomplete.com/presentations/logstash-monitorama-
↪2013/",
        "http_status": "200",
        "http_user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1)␣
↪AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36",
        "http_version": "HTTP/1.1",
        "identd": "-",
        "record_id": "4ca6a8b5-1a60-421e-9ae9-6c30330e497e",
        "record_raw_value": "83.149.9.216 - - [17/May/2015:10:05:43 +0000] \"GET /
↪presentations/logstash-monitorama-2013/images/kibana-dashboard3.png HTTP/1.1\" 200␣
↪171717 \"http://semicomplete.com/presentations/logstash-monitorama-2013/\" \
↪"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like␣
↪Gecko) Chrome/32.0.1700.77 Safari/537.36\"",
        "record_time": 14318571.4.10,
        "record_type": "apache_log",
        "src_ip": "83.149.9.216",
        "user": "-"
    }
}
```

You can now browse your data in Kibana and build great dashboards

1.2.12 Time series sampling & Outliers detection

In the following tutorial we'll handle time series data from a sensor. We'll see how sample the datapoints in a visually non destructive way and

We assume that you are already familiar with logisland platform and that you have successfully done the previous tutorials.

Note: You can download the latest release of logisland and the YAML configuration file for this tutorial which can be also found under *$LOGISLAND_HOME/conf* directory.

---

1. Setup the time series collection Stream

The first Stream use a KafkaRecordStreamParallelProcessing and chain of a SplitText

The first Processor simply parse the csv lines while the second index them into the search engine. Please note the output schema.

```
# parsing time series
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that links
```

(continues on next page)

(continued from previous page)

```
  configuration:
    kafka.input.topics: logisland_ts_raw
    kafka.output.topics: logisland_ts_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    avro.output.schema: >
      {  "version":1,
         "type": "record",
         "name": "com.hurence.logisland.record.cpu_usage",
         "fields": [
           { "name": "record_errors",   "type": [ {"type": "array", "items": "string"}
↪,"null"] },
           { "name": "record_raw_key", "type": ["string","null"] },
           { "name": "record_raw_value", "type": ["string","null"] },
           { "name": "record_id",    "type": ["string"] },
           { "name": "record_time", "type": ["long"] },
           { "name": "record_type", "type": ["string"] },
           { "name": "record_value",      "type": ["string","null"] } ]}
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
    - processor: apache_parser
      component: com.hurence.logisland.processor.SplitText
      type: parser
      documentation: a parser that produce events from an apache log REGEX
      configuration:
        record.type: apache_log
        value.regex: (\S+),(\S+)
        value.fields: record_time,record_value
```

2. Setup the Outliers detection Stream

The first Stream use a KafkaRecordStreamParallelProcessing and a DetectOutliers Processor

---

Note: It's important to see that we perform outliers detection in parallel. So if we would perform this detection for a particular grouping of record we would have used a KafkaRecordStreamSQLAggregator with a `GROUP BY` clause instead.

```
# detect outliers
- stream: detect_outliers
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that match query in parrallel
  configuration:
    kafka.input.topics: logisland_sensor_events
    kafka.output.topics: logisland_sensor_outliers_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
```

(continues on next page)

(continued from previous page)

```
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
    - processor: match_query
      component: com.hurence.logisland.processor.DetectOutliers
      type: processor
      documentation: a processor that detection something exotic in a continuous time␣
↪series values
      configuration:
        rotation.policy.type: by_amount
        rotation.policy.amount: 100
        rotation.policy.unit: points
        chunking.policy.type: by_amount
        chunking.policy.amount: 10
        chunking.policy.unit: points
        global.statistics.min: -100000
        min.amount.to.predict: 100
        zscore.cutoffs.normal: 3.5
        zscore.cutoffs.moderate: 5
        record.value.field: record_value
        record.time.field: record_time
        output.record.type: sensor_outlier
```

3. Setup the time series Sampling Stream

The first Stream use a KafkaRecordStreamParallelProcessing and a RecordSampler Processor

```
# sample time series
- stream: detect_outliers
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that match query in parrallel
  configuration:
    kafka.input.topics: logisland_sensor_events
    kafka.output.topics: logisland_sensor_sampled_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
    - processor: sampler
      component: com.hurence.logisland.processor.SampleRecords
      type: processor
      documentation: a processor that reduce the number of time series values
      configuration:
```

(continues on next page)

(continued from previous page)

```
        record.value.field: record_value
        record.time.field: record_time
        sampling.algorithm: average
        sampling.parameter: 10
```

4. Setup the indexing Stream

The last Stream use a KafkaRecordStreamParallelProcessing and chain of a SplitText and a BulkAddElasticsearch for indexing the whole records

```
# index records
- stream: indexing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that links
  configuration:
    kafka.input.topics: logisland_sensor_events,logisland_sensor_outliers_events,
↪logisland_sensor_sampled_events
    kafka.output.topics: none
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: none
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
    - processor: es_publisher
      component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
      type: processor
      documentation: a processor that trace the processed events
      configuration:
        elasticsearch.client.service: elasticsearch_service
        default.index: logisland
        default.type: event
        timebased.index: yesterday
        es.index.field: search_index
        es.type.field: record_type
```

4. Start logisland application

Connect a shell to your logisland container to launch the following stream processing job previously defined.

```
docker exec -ti logisland bash

#launch logisland streams
cd $LOGISLAND_HOME
bin/logisland.sh --conf conf/outlier-detection.yml

# send logs to kafka
cat cpu_utilization_asg_misconfiguration.csv | kafkacat -b sandbox:9092 -P -t␣
↪logisland_sensor_raw
```

(continues on next page)

(continued from previous page)

5. Check your alerts with Kibana

1.2.13 Bro/Logisland integration - Indexing Bro events

Bro and Logisland

Bro is a Network IDS (Intrusion Detection System) that can be deployed to monitor your infrastructure. Bro listens to the packets of your network and generates high level events from them. It can for instance generate an event each time there is a connection, a file transfer, a DNS query... anything that can be deduced from packet analysis.

Through its out-of-the-box ParseBroEvent processor, Logisland integrates with Bro and is able to receive and handle Bro events and notices coming from Bro. By analyzing those events with Logisland, you may do some correlations and for instance generate some higher level alarms or do whatever you want, in a scalable manner, like monitoring a huge infrastructure with hundreds of machines.

Bro comes with a scripting language that allows to also generate some higher level events from other events correlations. Bro calls such events 'notices'. For instance a notice can be generated when a user or bot tries to guess a password with brute forcing. Logisland is also able to receive and handle those notices.

For the purpose of this tutorial, we will show you how to receive Bro events and notices in Logisland and how to index them in ElasticSearch for network audit purpose. But you can imagine to plug any Logisland processors after the ParseBroEvent processor to build your own monitoring system or any other application based on Bro events and notices handling.

Tutorial environment

This tutorial will give you a better understanding of how Bro and Logisland integrate together.

We will start two Docker containers:

- 1 container hosting all the LogIsland services
- 1 container hosting Bro pre-loaded with Bro-Kafka plugin

We will launch two streaming processes and configure Bro to send events and notices to the Logisland system so that they are indexed in ElasticSearch.

It is important to understand that in a production environment Bro would be installed on machines where he is relevant for your infrastructure and will be configured to remotely point to the Logisland service (Kafka). But for easiness of this tutorial, we provide you with an easy mean of generating Bro events through our Bro Docker image.

This tutorial will guide you through the process of configuring Logisland for treating Bro events, and configuring Bro of the second container to send the events and notices to the Logisland service in the first container.

Note: You can download the latest release of Logisland and the YAML configuration file for this tutorial which can be also found under *$LOGISLAND_HOME/conf* directory in the Logsiland container.

---

1. Start the Docker container with LogIsland

LogIsland is packaged as a Docker image that you can build yourself or pull from Docker Hub. The docker image is built from a CentOs image with the following components already installed (among some others not useful for this

tutorial):

- Kafka
- Spark
- Elasticsearch
- LogIsland

Pull the image from Docker Repository (it may take some time)

```
docker pull hurence/logisland
```

You should be aware that this Docker container is quite eager in RAM and will need at least 8G of memory to run smoothly. Now run the container

```
# run container
docker run \
    -it \
    -p 80:80 \
    -p 8080:8080 \
    -p 3000:3000 \
    -p 9200-9300:9200-9300 \
    -p 5601:5601 \
    -p 2181:2181 \
    -p 9092:9092 \
    -p 9000:9000 \
    -p 4050-4060:4050-4060 \
    --name logisland \
    -h sandbox \
    hurence/logisland bash

# get container ip
docker inspect logisland | grep IPAddress

# or if your are on mac os
docker-machine ip default
```

You should add an entry for **sandbox** (with the container ip) in your `/etc/hosts` as it will be easier to access to all web services in Logisland running container. Or you can use 'localhost' instead of 'sandbox' where applicable.

Note: If you have your own Spark and Kafka cluster, you can download the latest release and unzip on an edge node.

2.Install required components

For this tutorial please make sure to already have installed elasticsearch and excel modules.

If not you can just do it through the components.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_2_4_0-
↪client:1.1.2
```

3. Transform Bro events into Logisland records

For this tutorial we will receive Bro events and notices and send them to Elastiscearch. The configuration file for this tutorial is already present in the container at `$LOGISLAND_HOME/conf/index-bro-events.yml` and its content can be viewed here . Within the following steps, we will go through this configuration file and detail the sections and what they do.

Connect a shell to your Logisland container to launch a Logisland instance with the following streaming jobs:

```
docker exec -ti logisland bash
cd $LOGISLAND_HOME
bin/logisland.sh --conf conf/index-bro-events.yml
```

Note: Logisland is now started. If you want to go straight forward and do not care for the moment about the configuration file details, you can now skip the following sections and directly go to the *4. Start the Docker container with Bro* section.

Setup Spark/Kafka streaming engine

An Engine is needed to handle the stream processing. The `conf/index-bro-events.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) as well as an Elasticsearch service that will be used later in the BulkAddElasticsearch processor.

```
engine:
  component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
  type: engine
  documentation: Index Bro events with LogIsland
  configuration:
    spark.app.name: IndexBroEventsDemo
    spark.master: local[4]
    spark.driver.memory: 1G
    spark.driver.cores: 1
    spark.executor.memory: 2G
    spark.executor.instances: 4
    spark.executor.cores: 2
    spark.yarn.queue: default
    spark.yarn.maxAppAttempts: 4
    spark.yarn.am.attemptFailuresValidityInterval: 1h
    spark.yarn.max.executor.failures: 20
    spark.yarn.executor.failuresValidityInterval: 1h
    spark.task.maxFailures: 8
    spark.serializer: org.apache.spark.serializer.KryoSerializer
    spark.streaming.batchDuration: 4000
    spark.streaming.backpressure.enabled: false
    spark.streaming.unpersist: false
    spark.streaming.blockInterval: 500
    spark.streaming.kafka.maxRatePerPartition: 3000
    spark.streaming.timeout: -1
    spark.streaming.unpersist: false
    spark.streaming.kafka.maxRetries: 3
    spark.streaming.ui.retainedBatches: 200
    spark.streaming.receiver.writeAheadLog.enable: false
    spark.ui.port: 4050
```

(continues on next page)

(continued from previous page)

```
  controllerServiceConfigurations:

    - controllerService: elasticsearch_service
      component: com.hurence.logisland.service.elasticsearch.Elasticsearch_2_4_0_
↪ClientService
      type: service
      documentation: elasticsearch 2.4.0 service implementation
      configuration:
        hosts: sandbox:9300
        cluster.name: elasticsearch
        batch.size: 20000

  streamConfigurations:
```

Stream 1: Parse incoming Bro events

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the Bro events and notices sent in the `bro` topic and push the processing output into the `logisland_events` topic.

```
# Parsing
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: A processor chain that transforms Bro events into Logisland records
  configuration:
    kafka.input.topics: bro
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
```

Within this stream there is a single processor in the processor chain: the `Bro` processor. It takes an incoming Bro event/notice JSON document and computes a Logisland `Record` as a sequence of fields that were contained in the JSON document.

```
# Transform Bro events into Logisland records
- processor: Bro adaptor
  component: com.hurence.logisland.processor.bro.ParseBroEvent
  type: parser
  documentation: A processor that transforms Bro events into LogIsland events
```

This stream will process Bro events as soon as they will be queued into the `bro` Kafka topic. Each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

Stream 2: Index the processed records into Elasticsearch

The second Kafka stream will handle `Records` pushed into the `logisland_events` topic to index them into ElasticSearch. So there is no need to define an output topic. The input topic is enough:

```
# Indexing
- stream: indexing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: processor
  documentation: A processor chain that pushes bro events to ES
  configuration:
    kafka.input.topics: logisland_events
    kafka.output.topics: none
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: none
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
```

The only processor in the processor chain of this stream is the `BulkAddElasticsearch` processor.

```
# Bulk add into ElasticSearch
- processor: ES Publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: A processor that pushes Bro events into ES
  configuration:
    elasticsearch.client.service: elasticsearch_service
    default.index: bro
    default.type: events
    timebased.index: today
    es.index.field: search_index
    es.type.field: record_type
```

The `default.index: bro` configuration parameter tells the processor to index events into an index starting with the `bro` string. The `timebased.index: today` configuration parameter tells the processor to use the current date after the index prefix. Thus the index name is of the form `/bro.2017.02.23`.

Finally, the `es.type.field: record_type` configuration parameter tells the processor to use the record field `record_type` of the incoming record to determine the ElasticSearch type to use within the index.

We will come back to these settings and what they do in the section where we see examples of events to illustrate the workflow.

4. Start the Docker container with Bro

For this tutorial, we provide Bro as a Docker image that you can build yourself or pull from Docker Hub. The docker image is built from an Ubuntu image with the following components already installed:

- Bro

- Bro-Kafka plugin

Note: Due to the fact that Bro requires a Kafka plugin to be able to send events to Kafka and that building the Bro-Kafka plugin requires some substantial steps (need Bro sources), for this tutorial, we are only focusing on configuring Bro, and consider it already compiled and installed with its Bro-Kafka plugin (this is the case in our Bro docker image). But looking at the Dockerfile we made to build the Bro Docker image and which is located here, you will have an idea on how to install Bro and Bro-Kafka plugin binaries on your own systems.

Pull the Bro image from Docker Repository:

Warning: If the Bro image is not yet available in the Docker Hub: please build our Bro Docker image yourself as described in the link above for the moment.

```
docker pull hurence/bro
```

Start a Bro container from the Bro image:

```
# run container
docker run -it --name bro -h bro hurence/bro

# get container ip
docker inspect bro | grep IPAddress

# or if your are on mac os
docker-machine ip default
```

5. Configure Bro to send events to Kafka

In the following steps, if you want a new shell to your running bro container, do as necessary:

```
docker exec -ti bro bash
```

Make the sandbox hostname reachable

Kafka in the Logisland container broadcasts his hostname which we have set up being `sandbox`. For this hostname to be reachable from the Bro container, we must declare the IP address of the Logisland container. In the Bro container, edit the `/etc/hosts` file and add the following line at the end of the file, using the right IP address:

```
172.17.0.2  sandbox
```

Note: Be sure to use the IP address of your Logisland container.

Note: Any potential communication problem of the Bro-Kafka plugin will be displayed in the `/usr/local/bro/spool/bro/stderr.log` log file. Also, you should not need this, but the advertised name used by Kafka is declared in the `/usr/local/kafka/config/server.properties` file (in the Logisland container), in the `advertised.host.name` property. Any modification to this property requires a Kafka server restart.

Edit the Bro config file

We will configure Bro so that it loads the Bro-Kafka plugin at startup. We will also point to Kafka of the Logisland container and define the event types we want to push to Logisland.

Edit the config file of bro:

```
vi $BRO_HOME/share/bro/site/local.bro
```

At the beginning of the file, add the following section (take care to respect indentation):

```
@load Bro/Kafka/logs-to-kafka.bro
    redef Kafka::kafka_conf = table(
        ["metadata.broker.list"] = "sandbox:9092",
        ["client.id"] = "bro"
    );
    redef Kafka::topic_name = "bro";
    redef Kafka::logs_to_send = set(Conn::LOG, DNS::LOG, SSH::LOG, Notice::LOG);
    redef Kafka::tag_json = T;
```

Let's detail a bit what we did:

This line tells Bro to load the Bro-Kafka plugin at startup (the next lines are configuration for the Bro-Kafka plugin):

```
@load Bro/Kafka/logs-to-kafka.bro
```

These lines make the Bro-Kafka plugin point to the Kafka instance in the Logisland container (host, port, client id to use). These are communication settings:

```
redef Kafka::kafka_conf = table(
    ["metadata.broker.list"] = "sandbox:9092",
    ["client.id"] = "bro"
    );
```

This line tells the Kafka topic name to use. It is important that it is the same as the input topic of the ParseBroEvent processor in Logisland:

```
redef Kafka::topic_name = "bro";
```

This line tells the Bro-Kafka plugin what type of events should be intercepted and sent to Kafka. For this tutorial we send Connections, DNS and SSH events. We are also interested in any notice (alert) that Bro can generate. For a complete list of possibilities, see the Bro documentation for events and notices. If you want all possible events and notices available by default to be sent into Kafka, just comment this line:

```
redef Kafka::logs_to_send = set(Conn::LOG, DNS::LOG, SSH::LOG, Notice::LOG);
```

This line tells the Bro-Kafka plugin to add the event type in the Bro JSON document it sends. This is required and expected by the Bro Processor as it uses this field to tag the record with his type. This also tells Logisland which ElasticSearch index type to use for storing the event:

```
redef Kafka::tag_json = T;
```

Start Bro

To start bro, we use the `broctl` command that is already in the path of the container. It starts an interactive session to control bro:

```
broctl
```

Then start the bro service: use the `deploy` command in broctl session:

```
Welcome to BroControl 1.5-9

Type "help" for help.

[BroControl] > deploy
checking configurations ...
installing ...
removing old policies in /usr/local/bro/spool/installed-scripts-do-not-touch/site ...
removing old policies in /usr/local/bro/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.bro ...
generating local-networks.bro ...
generating broctl-config.bro ...
generating broctl-config.sh ...
stopping ...
bro not running
starting ...
starting bro ...
```

Note: The `deploy` command is a shortcut to the `check`, `install` and `restart` commands. Everytime you modify the `$BRO_HOME/share/bro/site/local.bro` configuration file, you must re-issue a `deploy` command so that changes are taken into account.

6. Generate some Bro events and notices

Now that everything is in place you can generate some network activity in the Bro container to generate some events and see them indexed in ElasticSearch.

Monitor Kafka topic

We will generate some events but first we want to see them in Kafka to be sure Bro has forwarded them to Kafka. Connect to the Logisland container:

```
docker exec -ti logisland bash
```

Then use the `kafkacat` command to listen to messages incoming in the `bro` topic:

```
kafkacat -b localhost:9092 -t bro -o end
```

Let the command run. From now on, any incoming event from Bro and entering Kafka will be also displayed in this shell.

Issue a DNS query

Open a shell to the Bro container:

```
docker exec -ti bro bash
```

Then use the `ping` command to trigger an underlying DNS query:

```
ping www.wikipedia.org
```

You should see in the listening `kafkacat` shell an incoming JSON Bro event of type `dns`.

Here is a pretty print version of this event. It should look like this one:


```
{
  "dns": {
    "AA": false,
    "TTLs": [599],
    "id.resp_p": 53,
    "rejected": false,
    "query": "www.wikipedia.org",
    "answers": ["91.198.174.192"],
    "trans_id": 56307,
    "rcode": 0,
    "id.orig_p": 60606,
    "rcode_name": "NOERROR",
    "TC": false,
    "RA": true,
    "uid": "CJkHd3UABb4W7mx8b",
    "RD": false,
    "id.orig_h": "172.17.0.2",
    "proto": "udp",
    "id.resp_h": "8.8.8.8",
    "Z": 0,
    "ts": 1487785523.12837
  }
}
```


The Bro Processor should have processed this event which should have been handled next by the BulkAddElasticsearch processor and finally the event should have been stored in ElasticSearch in the Logisland container.

To see this stored event, we will query ElasticSearch with the `curl` command. Let's browse the `dns` type in any index starting with `bro`:

```
curl http://sandbox:9200/bro*/dns/_search?pretty
```

Note: Do not forget to change sandbox with the IP address of the Logisland container if needed.

You should be able to localize in the response from ElasticSearch a DNS event which looks like:


```
{
  "_index" : "bro.2017.02.23",
  "_type" : "dns",
  "_id" : "6aecfa3a-6a9e-4911-a869-b4e4599a69c1",
  "_score" : 1.0,
  "_source" : {
    "@timestamp": "2017-02-23T17:45:36Z",
    "AA": false,
    "RA": true,
    "RD": false,
```


(continues on next page)

(continued from previous page)


```
    "TC": false,
    "TTLs": [599],
    "Z": 0,
    "answers": ["91.198.174.192"],
    "id_orig_h": "172.17.0.2",
    "id_orig_p": 60606,
    "id_resp_h": "8.8.8.8",
    "id_resp_p": 53,
    "proto": "udp",
    "query": "www.wikipedia.org",
    "rcode": 0,
    "rcode_name": "NOERROR",
    "record_id": "1947d1de-a65e-42aa-982f-33e9c66bfe26",
    "record_time": 1487785536027,
    "record_type": "dns",
    "rejected": false,
    "trans_id": 56307,
    "ts": 1487785523.12837,
    "uid": "CJkHd3UABb4W7mx8b"
  }
}
```


You should see that this JSON document is stored in a indexed of the form `/bro.XXXX.XX.XX/dns`:

```
"_index" : "bro.2017.02.23",
"_type" : "dns",
```

Here, as the Bro event is of type `dns`, the event has been indexed using the `dns` ES type in the index. This allows to easily search only among events of a particular type.

The ParseBroEvent processor has used the first level field `dns` of the incoming JSON event from Bro to add a `record_type` field to the record he has created. This field has been used by the BulkAddElasticsearch processor to determine the index type to use for storing the record.

The `@timestamp` field is added by the BulkAddElasticsearch processor before pushing the record into ES. Its value is derived from the `record_time` field which has been added with also the `record_id` field by Logisland when the event entered Logisland. The `ts` field is the Bro timestamp which is the time when the event was generated in the Bro system.

Other second level fields of the incoming JSON event from Bro have been set as first level fields in the record created by the Bro Processor. Also any field that had a "." chacracter has been transformed to use a "_" character. For instance the `id.orig_h` field has been renamed into `id_orig_h`.

That is basically all the job the Bro Processor does. It's a small adaptation layer for Bro events. Now if you look in the Bro documentation and know the Bro event format, you can be able to know the format of a matching record going out of the ParseBroEvent processor. You should then be able to write some Logsisland processors to handle any record going out of the Bro Processor.

Issue a Bro Notice

As a Bro notice is the result of analysis of many events, generating a real notice event with Bro is a bit more complicated if you want to generate it with real traffic. Fortunately, Bro has the ability to generate events also from `pcap` files. These are "*packect capture*" files. They hold the recording of a real network traffic. Bro analyzes the packets in those files and generate events as if he was listening to real traffic.

In the Bro container, we have preloaded some `pcap` files in the `$PCAP_HOME` directory. Go into this directory:

```
cd $PCAP_HOME
```

The `ssh.pcap` file in this directory is a capture of a network traffic in which there is some SSH traffic with an attempt to guess a user password. The analysis of such traffic generates a Bro `SSH::Password_Guessing` notice.

Let's launch the following command to make Bro analyze the packets in the `ssh.pcap` file and generate this notice:

```
bro -r ssh.pcap local
```

In your previous `kafkacat` shell, you should see some `ssh` events that represent the SSH traffic. You should also see a `notice` event like this one:

```
{
  "notice": {
    "ts":1320435875.879278,
    "note":"SSH::Password_Guessing",
    "msg":"172.16.238.1 appears to be guessing SSH passwords (seen in 30 connections).
↪",
    "sub":"Sampled servers:  172.16.238.136, 172.16.238.136, 172.16.238.136, 172.16.
↪238.136, 172.16.238.136",
    "src":"172.16.238.1",
    "peer_descr":"bro",
    "actions":["Notice::ACTION_LOG"],
    "suppress_for":3600.0,
    "dropped":false
  }
}
```

Then, like for the DNS event, it should also have been indexed in the `notice` index type in ElastiSearch. Browse documents in this type like this:

```
curl http://sandbox:9200/bro*/notice/_search?pretty
```

Note: Do not forget to change sandbox with the IP address of the Logisland container if needed.

In the response, you should see a notice event like this:

```
{
   "_index" : "bro.2017.02.23",
   "_type" : "notice",
   "_id" : "76ab556b-167d-4594-8ee8-b05594cab8fc",
   "_score" : 1.0,
   "_source" : {
     "@timestamp" : "2017-02-23T10:45:08Z",
     "actions" : [ "Notice::ACTION_LOG" ],
     "dropped" : false,
     "msg" : "172.16.238.1 appears to be guessing SSH passwords (seen in 30␣
↪connections).",
     "note" : "SSH::Password_Guessing",
     "peer_descr" : "bro",
     "record_id" : "76ab556b-167d-4594-8ee8-b05594cab8fc",
     "record_time" : 1487933108041,
     "record_type" : "notice",
     "src" : "172.16.238.1",
     "sub" : "Sampled servers:  172.16.238.136, 172.16.238.136, 172.16.238.136, 172.
↪16.238.136, 172.16.238.136",
```

(continues on next page)

(continued from previous page)

```
    "suppress_for" : 3600.0,
    "ts" : 1.4.1435875879278E9
  }
}
```

We are done with this first approach of Bro integration with LogIsland.

As we configured Bro to also send SSH and Connection events to Kafka, you can have a look at the matching generated events in ES by browsing the `ssh` and `conn` index types:

```
# Browse SSH events
curl http://sandbox:9200/bro*/ssh/_search?pretty
# Browse Connection events
curl http://sandbox:9200/bro*/conn/_search?pretty
```

If you wish, you can also add some additional event types to be sent to Kafka in the Bro config file and browse the matching indexed events in ES using the same kind of `curl` commands just by changing the type in the query (do not forget to re-deploy Bro after configuration file modifications).

1.2.14 Netflow/Logisland integration - Handling Netflow traffic

Netflow and Logisland

Netflow is a feature introduced on Cisco routers that provides the ability to collect IP network traffic. We can distinguish 2 components:

- Flow exporter: aggregates packets into flows and exports flow records (binary format) towards flow collectors
- Flow collector: responsible for reception, storage and pre-processing of flow data received from a flow exporter

The collected data are therefore available for analysis purpose (intrusion detection, traffic analysis...)

Network Flows: A network flow can be defined in many ways. Cisco standard NetFlow version 5 defines a flow as a unidirectional sequence of packets that all share the following 7 values:

1. Ingress interface (SNMP ifIndex)
2. Source IP address
3. Destination IP address
4. IP protocol
5. Source port for UDP or TCP, 0 for other protocols
6. Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols
7. IP Type of Service

NetFlow Record

A NetFlow record can contain a wide variety of information about the traffic in a given flow. NetFlow version 5 (one of the most commonly used versions, followed by version 9) contains the following:

- Input interface index used by SNMP (ifIndex in IF-MIB).
- Output interface index or zero if the packet is dropped.
- Timestamps for the flow start and finish time, in milliseconds since the last boot.
- Number of bytes and packets observed in the flow

- Layer 3 headers:
  - Source & destination IP addresses
  - ICMP Type and Code.
  - IP protocol
  - Type of Service (ToS) value
- Source and destination port numbers for TCP, UDP, SCTP
- For TCP flows, the union of all TCP flags observed over the life of the flow.
- Layer 3 Routing information:
  - IP address of the immediate next-hop (not the BGP nexthop) along the route to the destination
  - Source & destination IP masks (prefix lengths in the CIDR notation)

Through its out-of-the-box Netflow processor, Logisland integrates with Netflow (V5) and is able to receive and handle Netflow events coming from a netflow collector. By analyzing those events with Logisland, you may do some analysis for example for intrusion detection or traffic analysis.

In this tutorial, we will show you how to generate some Netflow traffic in LogIsland and how to index them in ElasticSearch and vizualize them in Kinbana. More complexe treatment could bv done by plugging any Logisland processors after the Netflow processor.

Tutorial environment

This tutorial aims to show how to handle Netflow traffic within LogIsland.

For the purpose of this tutorial, we will generate Netflow traffic using nfgen. This tool will simulate a netflow traffic and send binary netflow records on port 2055 of sandbox. A nifi instance running on sandbox will listen on that port for incoming traffic and push the binary events to a kafka broker.

We will launch two streaming processes, one for generating the corresponding Netflow LogIsland records and the second one to index them in ElasticSearch.

Note: It is important to understand that in real environment Netflow traffic will be triggered by network devices (router, switches,. . .), so you will have to get the netflow traffic from the defined collectors, and send the corresponding record (formatted in JSON format as described before) to the Logisland service (Kafka).

Note: You can download the latest release of Logisland and the YAML configuration file for this tutorial which can also be found under *$LOGISLAND_HOME/conf* directory in the LogIsland container.

---

1. Start LogIsland as a Docker container

LogIsland is packaged as a Docker container that you can build yourself or pull from Docker Hub. The docker container is built from a Centos 6.4 image with the following tools enabled (among others)

- Kafka
- Spark
- Elasticsearch
- Kibana

- LogIsland

Pull the image from Docker Repository (it may take some time)

```
docker pull hurence/logisland
```

You should be aware that this Docker container is quite eager in RAM and will need at least 8G of memory to run smoothly. Now run the container

```
# run container
docker run \
    -it \
    -p 80:80 \
    -p 8080:8080 \
    -p 2055:2055 \
    -p 3000:3000 \
    -p 9200-9300:9200-9300 \
    -p 5601:5601 \
    -p 2181:2181 \
    -p 9092:9092 \
    -p 9000:9000 \
    -p 4050-4060:4050-4060 \
    --name logisland \
    -h sandbox \
    hurence/logisland bash

# get container ip
docker inspect logisland

# or if your are on mac os
docker-machine ip default
```

you should add an entry for **sandbox** (with the container ip) in your `/etc/hosts` as it will be easier to access to all web services in logisland running container.

Note: If you have your own Spark and Kafka cluster, you can download the latest release and unzip on an edge node.

2. Configuration steps

First we have to peform some configuration steps on sandbox (to configure and start elasticsearch and nifi). We will create a dynamic template in ElasticSearch (to better handle the field mapping) using the following command:

```
docker exec -ti logisland bash

[root@sandbox /]# curl -XPUT localhost:9200/_template/netflow -d '{
  "template" : "netflow.*",
  "settings": {
    "index.refresh_interval": "5s"
  },
  "mappings" : {
    "netflowevent" : {
      "numeric_detection": true,
      "_all" : {"enabled" : false},
      "properties" : {
        "dOctets": {"index": "analyzed", "type": "long" },
```

(continues on next page)

(continued from previous page)

```
        "dPkts": { "index": "analyzed", "type": "long" },
        "dst_as": { "index": "analyzed", "type": "long" },
        "dst_mask": { "index": "analyzed", "type": "long" },
        "dst_ip4": { "index": "analyzed", "type": "ip" },
        "dst_port": { "index": "analyzed", "type": "long" },
        "first":{"index": "analyzed", "type": "long" },
        "input":{"index": "analyzed", "type": "long" },
        "last":{"index": "analyzed", "type": "long" },
        "nexthop":{"index": "analyzed", "type": "ip" },
        "output":{"index": "analyzed", "type": "long" },
        "nprot":{"index": "analyzed", "type": "long" },
        "record_time":{"index": "analyzed", "type": "date","format": "strict_date_
↪optional_time||epoch_millis" },
        "src_as":{"index": "analyzed", "type": "long" },
        "src_mask":{"index": "analyzed", "type": "long" },
        "src_ip4": { "index": "analyzed", "type": "ip" },
        "src_port":{"index": "analyzed", "type": "long" },
        "flags":{"index": "analyzed", "type": "long" },
        "tos":{"index": "analyzed", "type": "long" },
        "unix_nsecs":{"index": "analyzed", "type": "long" },
        "unix_secs":{"index": "analyzed", "type": "date","format": "strict_date_
↪optional_time||epoch_second" }
      }
    }
  }
}'
```

In order to send netflow V5 event (binary format) to `logisland_raw` Kafka topic, we will use a nifi instance which will simply listen for netflow traffic on a UDP port (we keep here the default netflow port 2055) and push these netflow records to a kafka broker (sandbox:9092 with topic `netflow`).

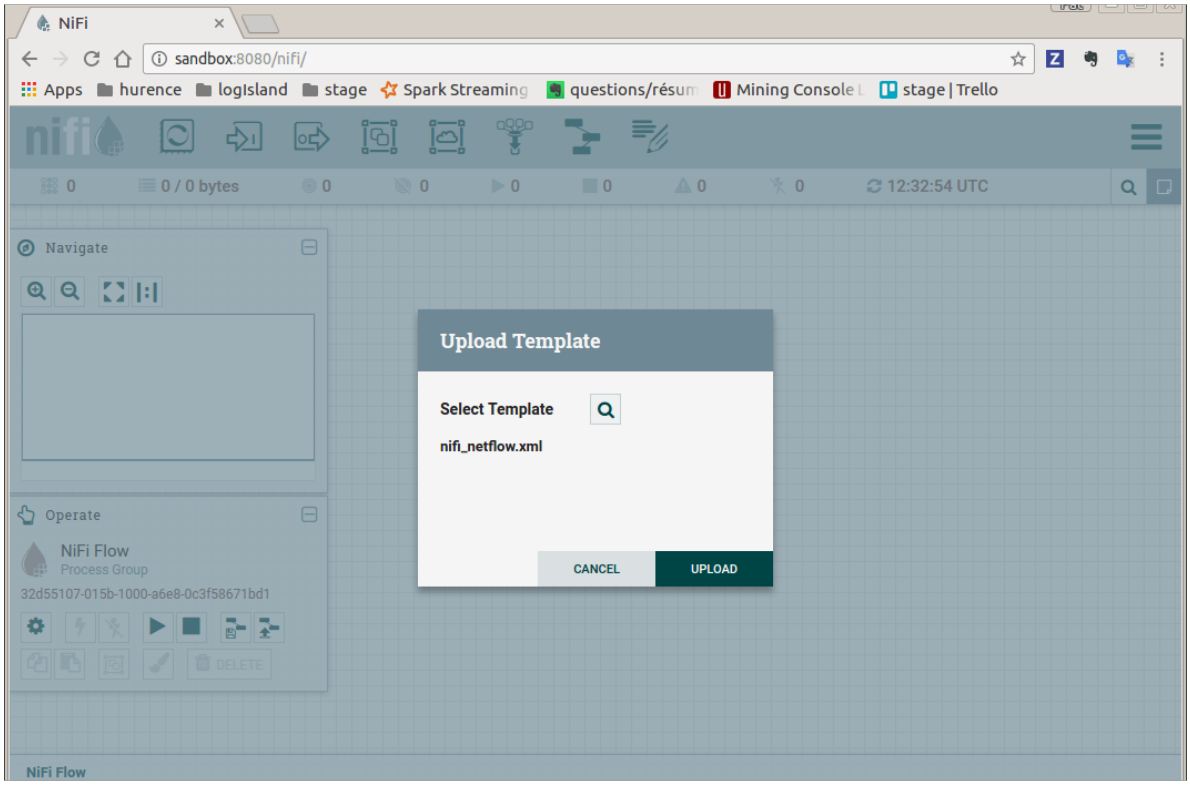
1. Start nifi

   ```
   docker exec -ti logisland bash
   cd /usr/local/nifi-1.1.2
   bin/nifi.sh start
   ```

   browse http://sandbox:8080/nifi/

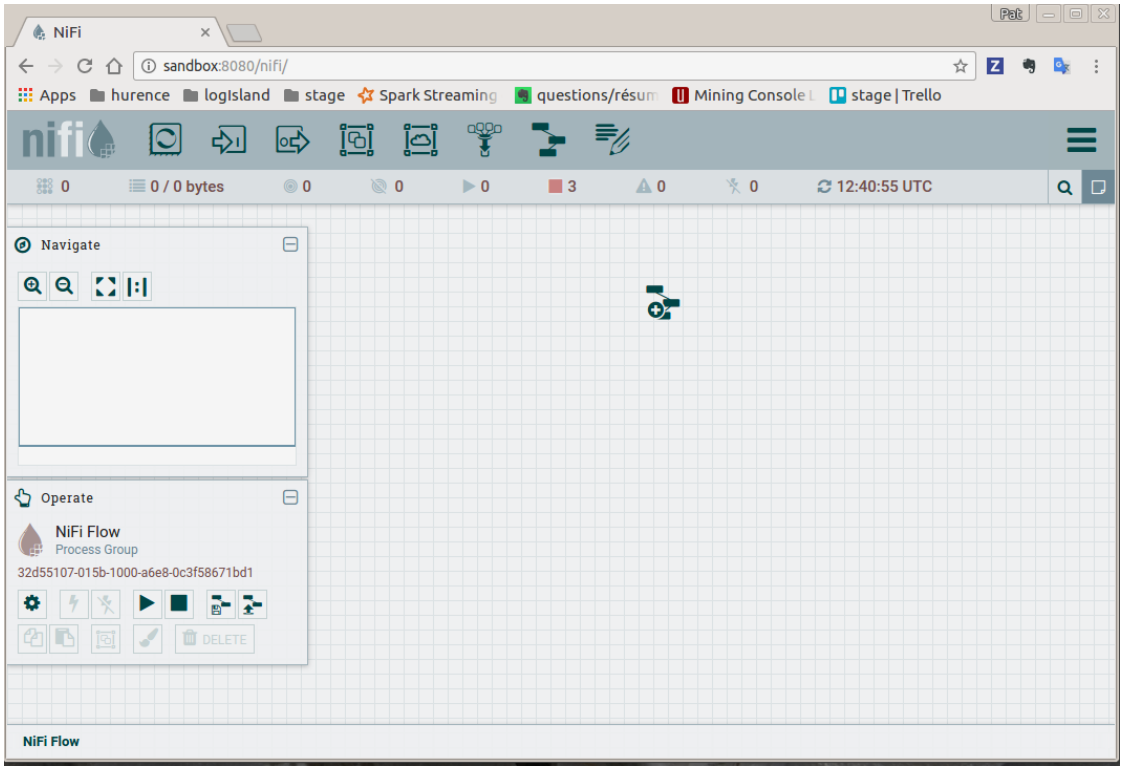
2. Import flow template

   Download this nifi template and import it using “Upload Template” in “Operator” toolbox.

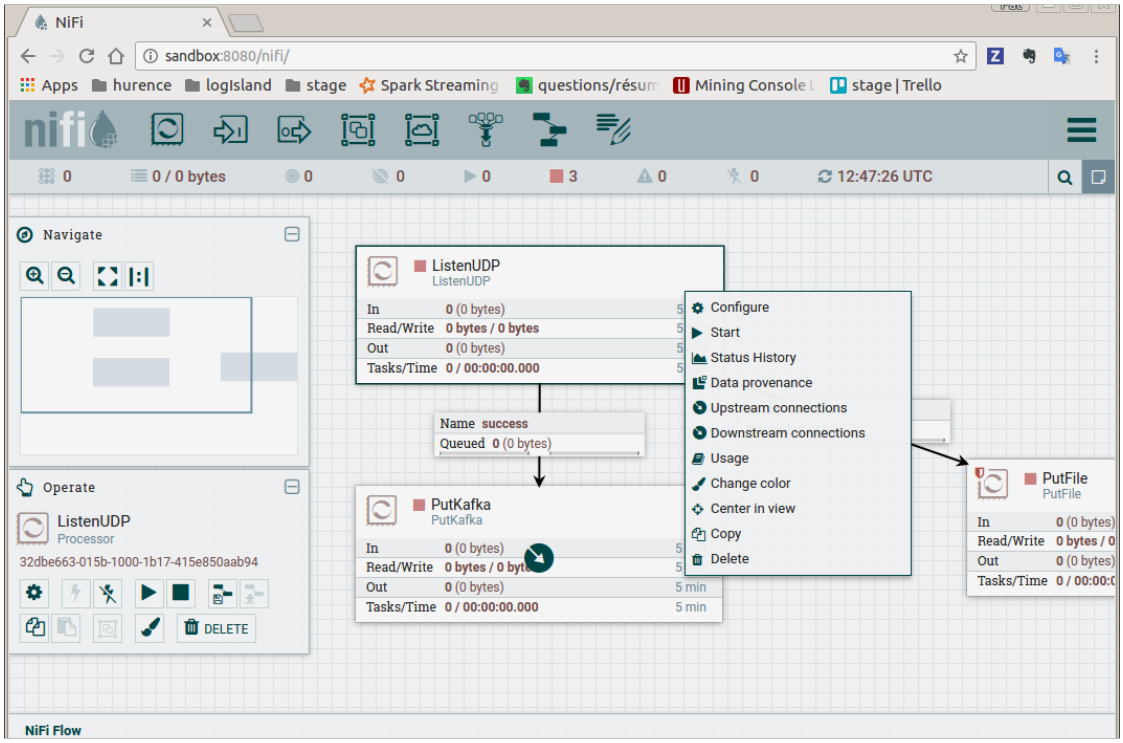


3. Use this template to create the nifi flow

   Drag the nifi toolbar template icon in the nifi work area and choose “nifi_netflow” template, the press “ADD” button



You finally have the following nifi flow



4. start nifi processors

   Select listenUDP processor of nifi flow, right click on it and press "Start". Do the same for putKafka processor.

Note: the PutFile processor is only for debugging purpose. It dumps netflow records to /tmp/netflow directory (that should be previously created). So you normally don't have to start it for that demo.

---

3. Parse Netflow records

For this tutorial we will handle netflow binary events, generate corresponding logisland records and store them to Elastiscearch

Connect a shell to your logisland container to launch the following streaming jobs.

```
docker exec -ti logisland bash
cd $LOGISLAND_HOME
bin/logisland.sh --conf conf/index-netflow-events.yml
```

Setup Spark/Kafka streaming engine

An Engine is needed to handle the stream processing. This `conf/index-netflow-events.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine (we will use a KafkaStream-ProcessingEngine) as well as an Elasticsearch service that will be used later in the BulkAddElasticsearch processor.

```
engine:
 component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
 type: engine
 documentation: Index Netflow events with LogIsland
 configuration:
   spark.app.name: IndexNetFlowEventsDemo
   spark.master: local[4]
   spark.driver.memory: 1G
   spark.driver.cores: 1
   spark.executor.memory: 2G
   spark.executor.instances: 4
   spark.executor.cores: 2
   spark.yarn.queue: default
   spark.yarn.maxAppAttempts: 4
   spark.yarn.am.attemptFailuresValidityInterval: 1h
   spark.yarn.max.executor.failures: 20
   spark.yarn.executor.failuresValidityInterval: 1h
   spark.task.maxFailures: 8
   spark.serializer: org.apache.spark.serializer.KryoSerializer
   spark.streaming.batchDuration: 4000
   spark.streaming.backpressure.enabled: false
   spark.streaming.unpersist: false
   spark.streaming.blockInterval: 500
   spark.streaming.kafka.maxRatePerPartition: 3000
   spark.streaming.timeout: -1
   spark.streaming.unpersist: false
   spark.streaming.kafka.maxRetries: 3
   spark.streaming.ui.retainedBatches: 200
   spark.streaming.receiver.writeAheadLog.enable: false
   spark.ui.port: 4050

 controllerServiceConfigurations:

   - controllerService: elasticsearch_service
     component: com.hurence.logisland.service.elasticsearch.Elasticsearch_2_4_0_
↪ClientService
     type: service
     documentation: elasticsearch 2.4.0 service implementation
     configuration:
       hosts: sandbox:9300
       cluster.name: elasticsearch
       batch.size: 20000

 streamConfigurations:
```

Stream 1 : parse incoming Netflow (Binary format) lines

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

We can define some serializers to marshall all records from and to a topic.

```
# Parsing
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
```

(continues on next page)

(continued from previous page)

```
  type: stream
  documentation: A processor chain that transforms Netflow events into Logisland␣
↪records
  configuration:
    kafka.input.topics: netflow
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 2
  processorConfigurations:
```

Within this stream there is a single processor in the processor chain: the Netflow processor. It takes an incoming Netflow event/notice binary record, parses it and computes a Logisland Record as a sequence of fields that were contained in the binary record.

```
# Transform Netflow events into Logisland records
     - processor: Netflow adaptor
       component: com.hurence.logisland.processor.netflow.ParseNetflowEvent
       type: parser
       documentation: A processor that transforms Netflow events into LogIsland events
       configuration:
         debug: false
         enrich.record: false
```

This stream will process log entries as soon as they will be queued into `logisland_raw` Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

Stream 2: Index the processed records into Elasticsearch

The second Kafka stream will handle `Records` pushed into the `logisland_events` topic to index them into ElasticSearch. So there is no need to define an output topic:

```
# Indexing
- stream: indexing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: processor
  documentation: A processor chain that pushes netflow events to ES
  configuration:
    kafka.input.topics: logisland_events
    kafka.output.topics: none
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: none
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
```

(continues on next page)

(continued from previous page)

```
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
```

The only processor in the processor chain of this stream is the `BulkAddElasticsearch` processor.

```
# Bulk add into ElasticSearch
- processor: ES Publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: A processor that pushes Netflow events into ES
  configuration:
    elasticsearch.client.service: elasticsearch_service
    default.index: netflow
    default.type: events
    timebased.index: today
    es.index.field: search_index
    es.type.field: record_type
```

The `default.index: netflow` configuration parameter tells the processor to index events into an index starting with the `netflow` string. The `timebased.index: today` configuration parameter tells the processor to use the current date after the index prefix. Thus the index name is of the form `/netflow.2017.03.30`.

Finally, the `es.type.field: record_type` configuration parameter tells the processor to use the record field `record_type` of the incoming record to determine the ElasticSearch type to use within the index.

4. Inject Netflow events into the system

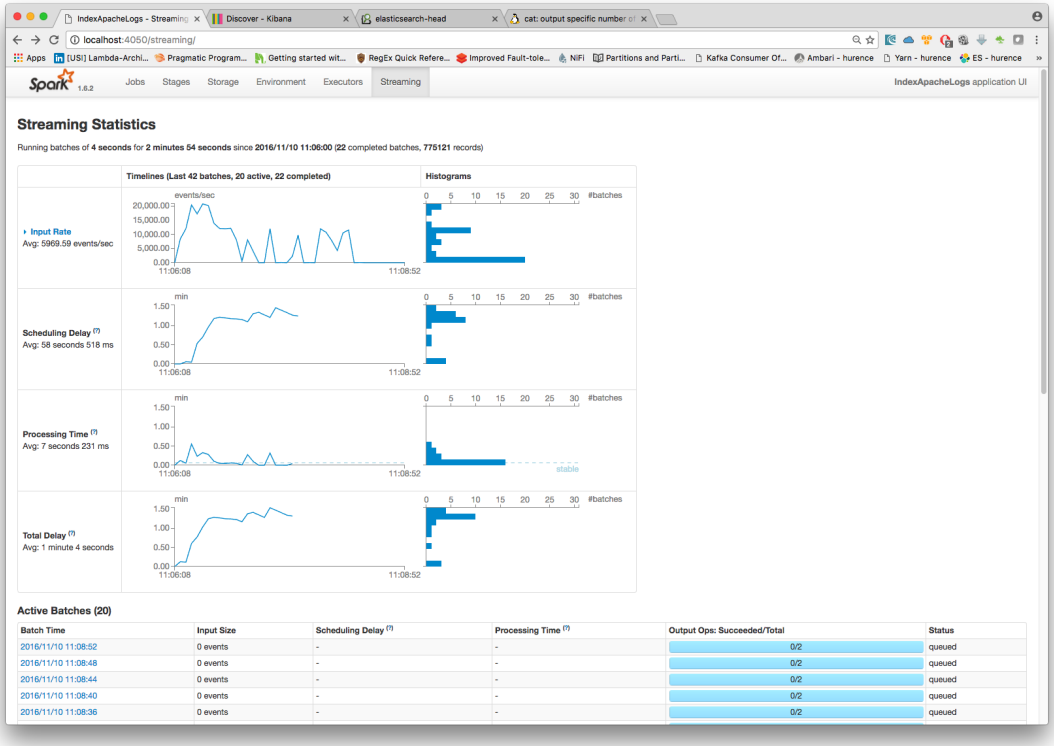
Generate Netflow events to port 2055 of localhost

Now that we have our nifi flow listening on port 2055 from Netflow (V5) traffic and push them to kafka, we have to generate netflow traffic. For the purpose of this tutorial, as already mentioned, we will install and use a netflow traffic generator (but you can use whatever other way to generate Netflow V5 traffic to port 2055)

```
docker exec -ti logisland bash
cd /tmp
wget https://github.com/pazdera/NetFlow-Exporter-Simulator/archive/master.zip
unzip master.zip
cd NetFlow-Exporter-Simulator-master/
make
./nfgen   #this command will generate netflow V5 traffic and send it to local port␣
↪2055.
```

5. Monitor your spark jobs and Kafka topics

Now go to http://sandbox:4050/streaming/ to see how fast Spark can process your data

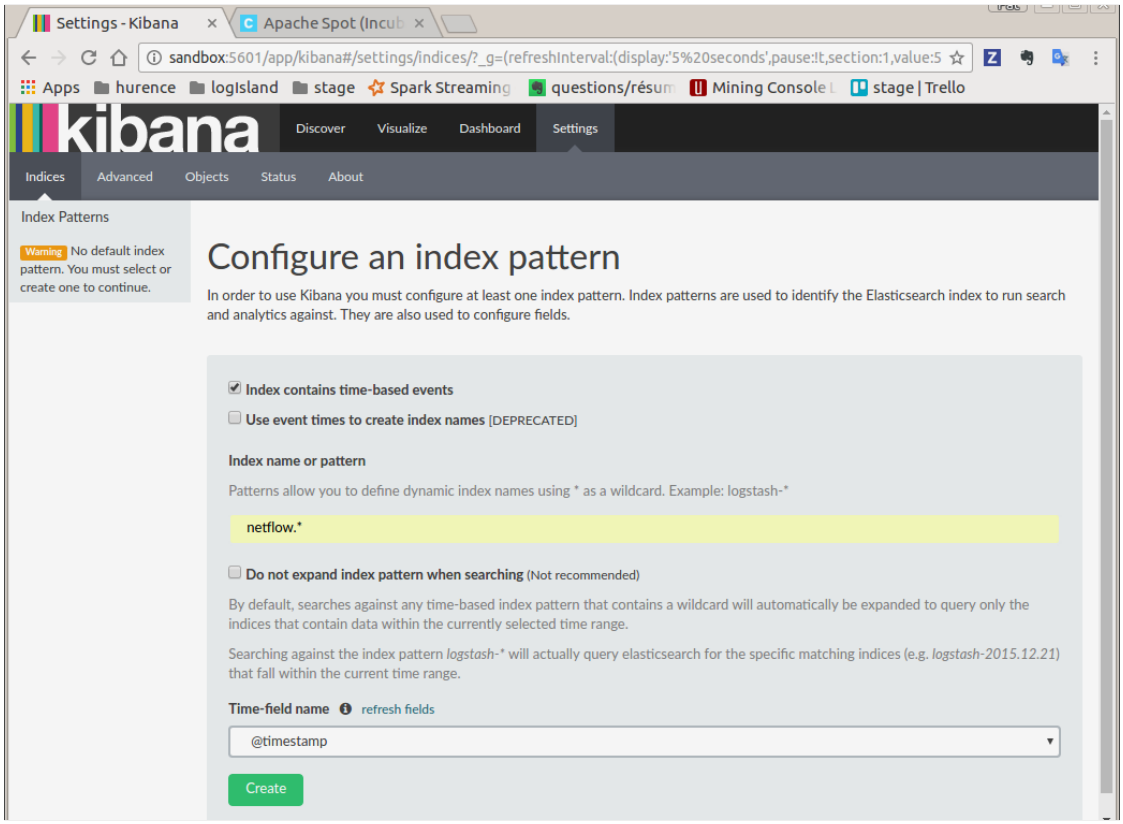


6. Use Kibana to inspect events

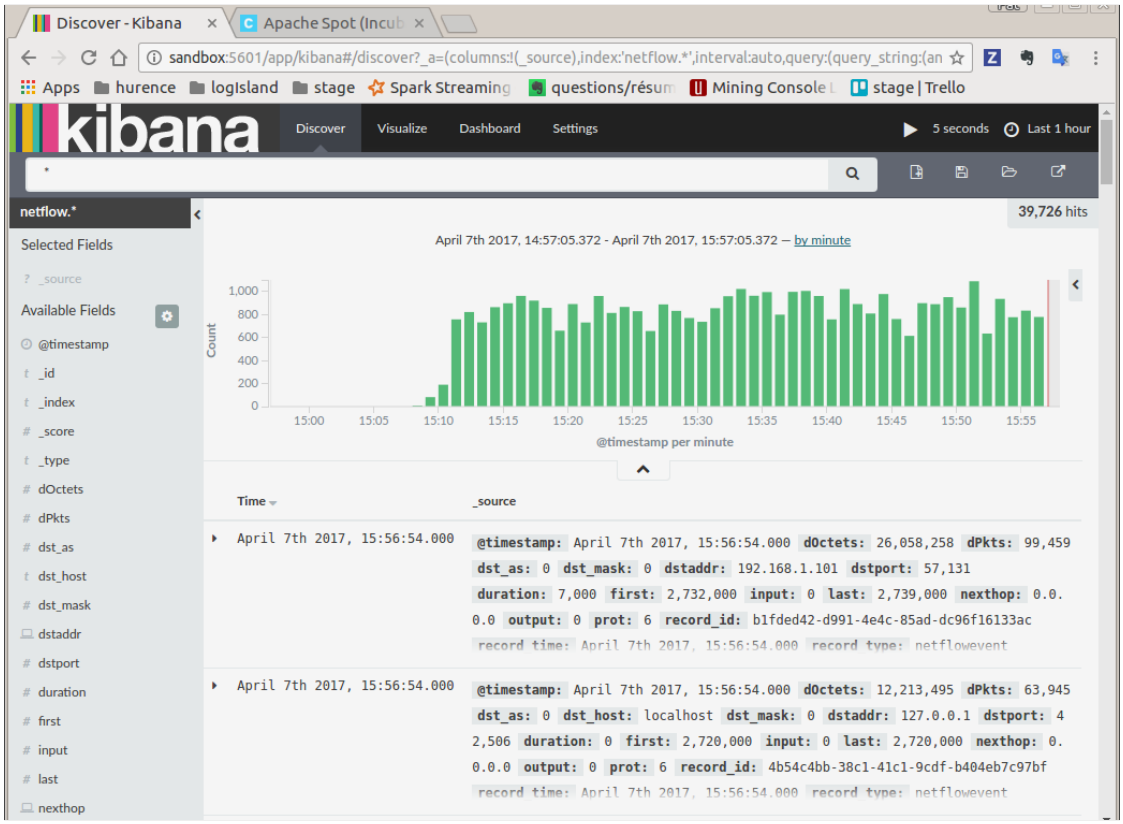
Inspect Netflow events under `Discover` tab

Open your browser and go to http://sandbox:5601/

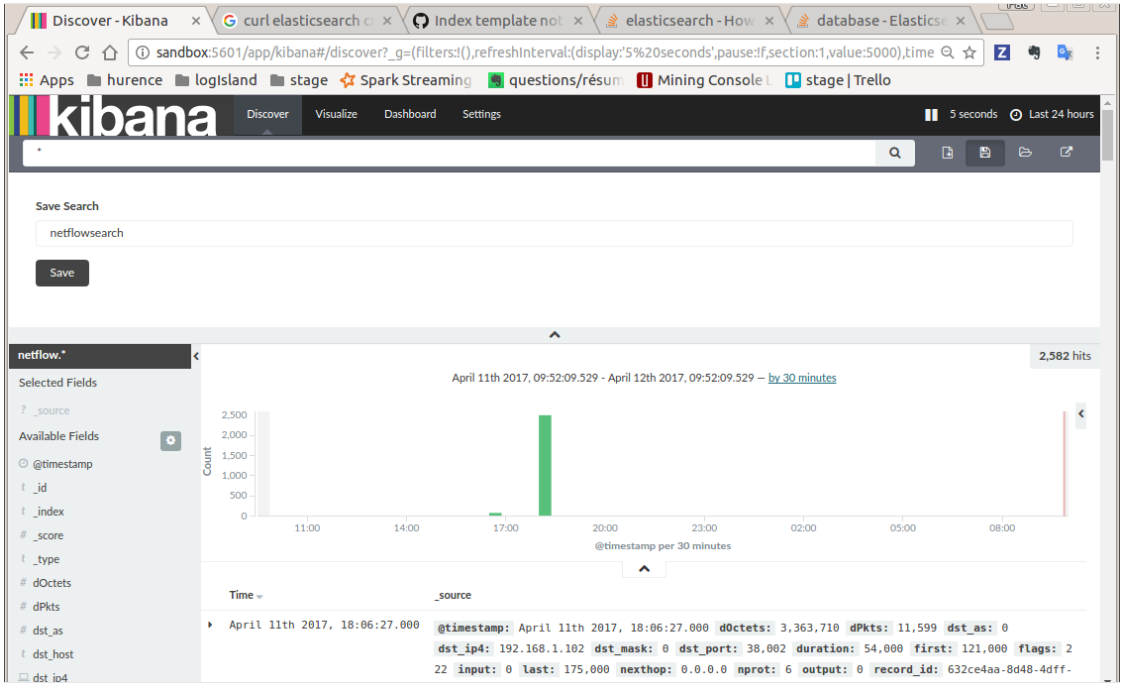
Configure a new index pattern with `netflow.*` as the pattern name and `@timestamp` as the time value field.



Then browse "Discover" tab, you should be able to explore your Netflow events.



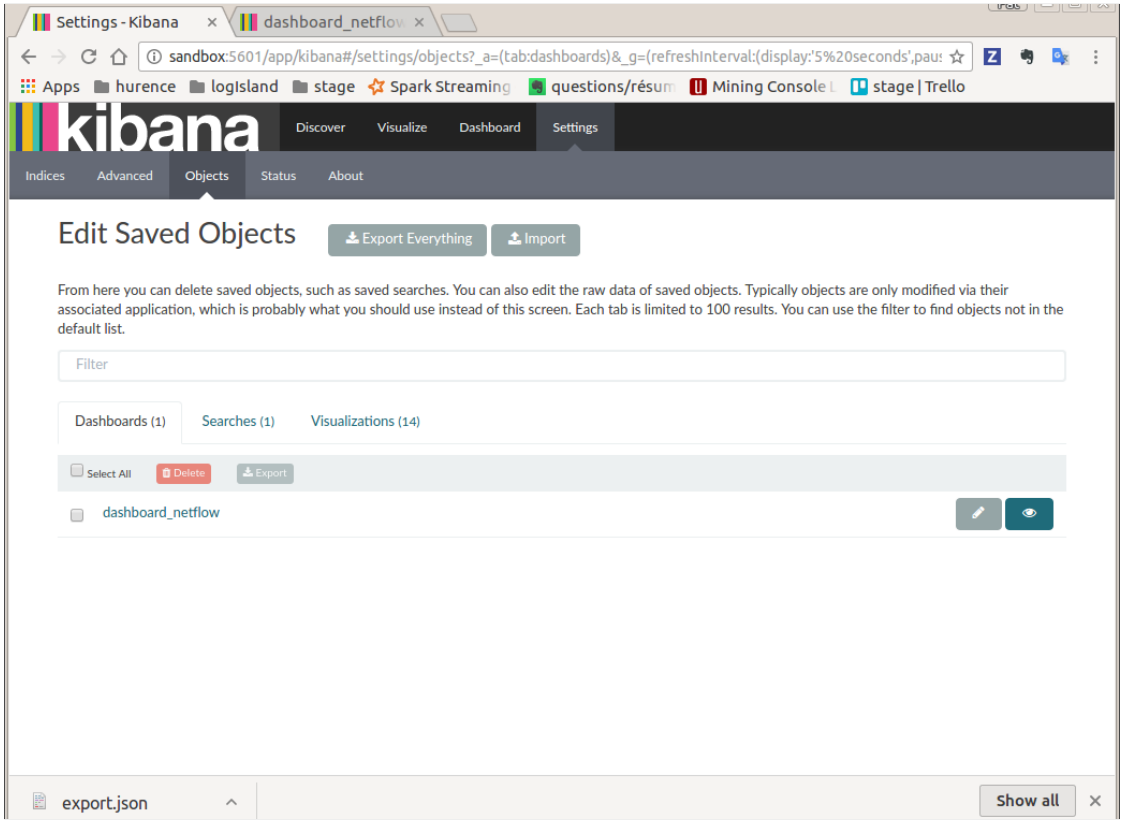
You have now to save your search by clicking the save icon. Save this search as "netflowsearch"



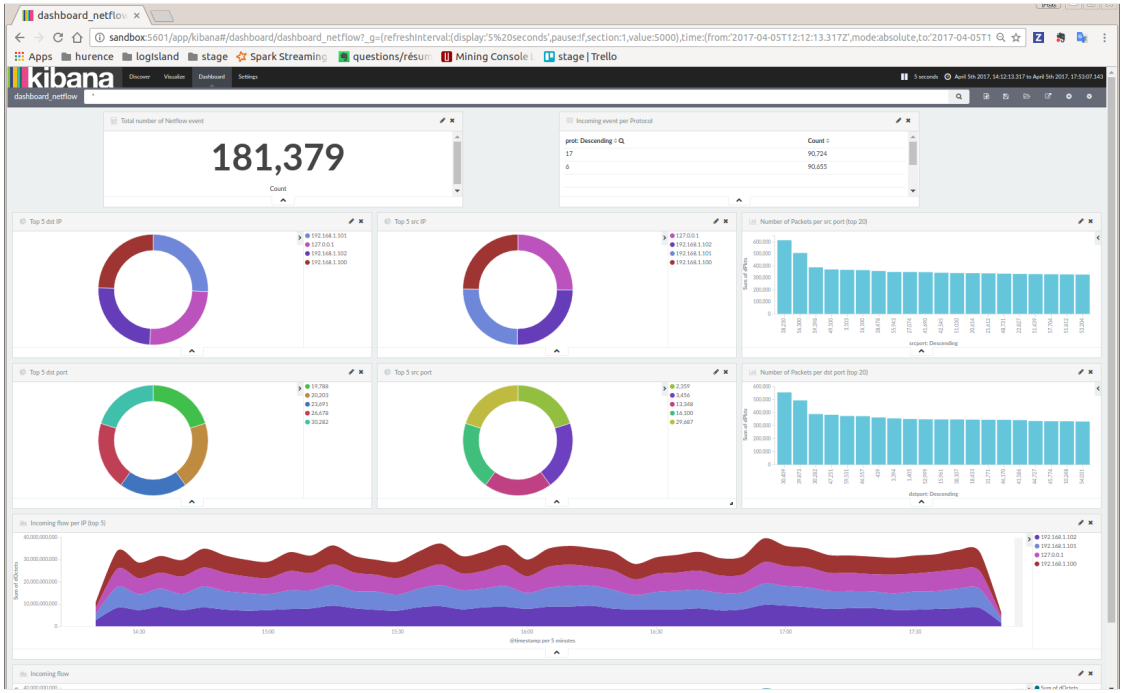
Display network information in kibana dashboard

First, you need to import the predefined Kibana dashboard (download this file first) under `Settings` tab, `Objetcs` subtab.

Select `Import` and load previously saved netflow_dashboard.json dashboard (it also contains required Kibana visualizations)



Then visit `Dashboard` tab, and open `dashboard_netflow` dashboard by clicking on `Load Saved Dashboard`. You should be able to visualize information about the generated traffic (choose the right time window, corresponding to the time of your traffic, in the right upper corner of kibana page)



1.2.15 Capturing Network packets in Logisland

1. Network Packets

A network packet is a formatted unit of data carried by a network from one computer (or device) to another. For example, a web page or an email are carried as a series of packets of a certain size in bytes. Each packet carries the information that will help it get to its destination : the sender's IP address, the intended receiver's IP address, something that tells the network how many packets the message has been broken into, . . .

Packet Headers

1. Protocol headers (IP, TCP, . . .)

This information is stored in different layers called "headers", encapsulating the packet payload. For example, a TCP/IP packet is wrapped in a TCP header, which is in turn encapsulated in an IP header.

The individual packets for a given file or message may travel different routes through the Internet. When they have all arrived, they are reassembled by the TCP layer at the receiving end.

2. PCAP format specific headers

Packets can be either analysed in real-time (stream mode) or stored in files for upcoming analysis (batch mode). In the latter case, the packets are stored in the pcap format, adding some specific headers :

- a global header is added in the beginning of the pcap file
- a packet header is added in front of each packet

In this tutorial we are going to **capture packets in live stream mode**

Why capturing network packets ?

Packet sniffing, or packet analysis, is the process of capturing any data transmitted over the local network and searching for any information that may be useful for :

- Troubleshooting network problems
- Detecting network intrusion attempts
- Detecting network misuse by internal and external users
- Monitoring network bandwidth utilization
- Monitoring network and endpoint security status
- Gathering and report network statistics

Packets information collected by Logisland

LogIsland parses all the fields of IP protocol headers, namely :

1. IP Header fields

- IP version : ip_version
- Internet Header Length : ip_internet_header_length
- Type of Service : ip_type_of_service
- Datagram Total Length : ip_datagram_total_length
- Identification : ip_identification
- Flags : ip_flags
- Fragment offset : ip_fragment_offset
- Time To Live : ip_time_to_live
- Protocol : protocol
- Header Checksum : ip_checksum
- Source IP address : src_ip
- Destination IP address : dst_ip
- Options : ip_options (variable size)
- Padding : ip_padding (variable size)

2. TCP Header fields

- Source port number : src_port
- Destination port number : dest_port
- Sequence Number : tcp_sequence_number
- Acknowledgment Number : tcp_acknowledgment_number
- Data offset : tcp_data_offset
- Flags : tcp_flags
- Window size : tcp_window_size
- Checksum : tcp_checksum

- Urgent Pointer : tcp_urgent_pointer
- Options : tcp_options (variable size)
- Padding : tcp_padding (variable size)

3. UDP Header fields

- Source port number : src_port
- Destination port number : dest_port
- Segment total length : udp_segment_total_length
- Checksum : udp_checksum

2. Tutorial environment

This tutorial aims to show how to capture live Network Packets and process then in LogIsland. Through its out-of-the-box ParseNetworkPacket processor, LogIsland is able to receive and handle network packets captured by a packet sniffer tool. Using LogIsland, you will be able to inspect those packets for network security, optimization or monitoring reasons.

In this tutorial, we will show you how to capture network packets, process those packets in LogIsland, index them in ElasticSearch and then display them in Kibana.

We will launch two streaming processors, one for parsing Network Packets into LogIsland packet records, and one to index those packet records in ElasticSearch.

Packet Capture Tool

For the purpose of this tutorial, we are going to capture network packets (off-the-wire) into a kafka topic using pycapa Apache probe, a tool based on Pcapy, a Python extension module that interfaces with the libpcap packet capture library.

For information, it is also possible to use the fastcapa Apache probe, based on DPDK, intended for high-volume packet capture.

Note: You can download the latest release of LogIsland and the YAML configuration file for this tutorial which can be also found under *$LOGISLAND_HOME/conf* directory in the LogIsland container.

---

3. Start LogIsland as a Docker container

LogIsland is packaged as a Docker container that you can build yourself or pull from Docker Hub. The docker container is built from a Centos 6.4 image with the following tools enabled (among others)

- Kafka
- Spark
- Elasticsearch
- Kibana
- LogIsland

Pull the image from Docker Repository (it may take some time)

```
docker pull hurence/logisland
```

You should be aware that this Docker container is quite eager in RAM and will need at least 8G of memory to run smoothly. Now run the container

```
# run container
docker run \
    -it \
    -p 80:80 \
    -p 8080:8080 \
    -p 3000:3000 \
    -p 9200-9300:9200-9300 \
    -p 5601:5601 \
    -p 2181:2181 \
    -p 9092:9092 \
    -p 9000:9000 \
    -p 4050-4060:4050-4060 \
    --name logisland \
    -h sandbox \
    hurence/logisland bash

# get container ip
docker inspect logisland

# or if your are on mac os
docker-machine ip default
```

you should add an entry for **sandbox** (with the container ip) in your `/etc/hosts` as it will be easier to access to all web services in logisland running container.

Note: If you have your own Spark and Kafka cluster, you can download the latest release and unzip on an edge node.

---

4. Parse Network Packets

In this tutorial we will capture network packets, process those packets in LogIsland and index them in ElasticSearch.

Connect a shell to your logisland container to launch LogIsland streaming jobs :

```
docker exec -ti logisland bash
cd $LOGISLAND_HOME
bin/logisland.sh --conf conf/index-network-packets.yml
```

Setup Spark/Kafka streaming engine

An Engine is needed to handle the stream processing. This `conf/index-network-packets.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine, we will use a KafkaStreamProcessingEngine :

```
engine:
 component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
 type: engine
 documentation: Parse network packets with LogIsland
```

(continues on next page)

(continued from previous page)

```
configuration:
  spark.app.name: ParseNetworkPacketDemo
  spark.master: local[4]
  spark.driver.memory: 1G
  spark.driver.cores: 1
  spark.executor.memory: 2G
  spark.executor.instances: 4
  spark.executor.cores: 2
  spark.yarn.queue: default
  spark.yarn.maxAppAttempts: 4
  spark.yarn.am.attemptFailuresValidityInterval: 1h
  spark.yarn.max.executor.failures: 20
  spark.yarn.executor.failuresValidityInterval: 1h
  spark.task.maxFailures: 8
  spark.serializer: org.apache.spark.serializer.KryoSerializer
  spark.streaming.batchDuration: 4000
  spark.streaming.backpressure.enabled: false
  spark.streaming.unpersist: false
  spark.streaming.blockInterval: 500
  spark.streaming.kafka.maxRatePerPartition: 3000
  spark.streaming.timeout: -1
  spark.streaming.unpersist: false
  spark.streaming.kafka.maxRetries: 3
  spark.streaming.ui.retainedBatches: 200
  spark.streaming.receiver.writeAheadLog.enable: false
  spark.ui.port: 4050

controllerServiceConfigurations:

  - controllerService: elasticsearch_service
    component: com.hurence.logisland.service.elasticsearch.Elasticsearch_2_4_0_
↪ClientService
    type: service
    documentation: elasticsearch 2.4.0 service implementation
    configuration:
      hosts: sandbox:9300
      cluster.name: elasticsearch
      batch.size: 4000

streamConfigurations:
```

Stream 1 : parse incoming Network Packets

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_input_packets_topic` topic and push the processed packet records into `logisland_parsed_packets_topic` topic.

We can define some serializers to marshall all records from and to a topic.

```
# Parsing
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: A processor chain that parses network packets into Logisland records
  configuration:
```

(continues on next page)

(continued from previous page)

```
    kafka.input.topics: logisland_input_packets_topic
    kafka.output.topics: logisland_parsed_packets_topic
    kafka.error.topics: logisland_error_packets_topic
    kafka.input.topics.serializer: com.hurence.logisland.serializer.
↪BytesArraySerializer
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
```

Within this stream there is a single processor in the processor chain: the ParseNetworkPacket processor. It takes an incoming network packet, parses it and computes a LogIsland record as a sequence of fields corresponding to packet headers fields.

```
# Transform network packets into LogIsland packet records
- processor: ParseNetworkPacket processor
  component: com.hurence.logisland.processor.networkpacket.ParseNetworkPacket
  type: parser
  documentation: A processor that parses network packets into LogIsland records
  configuration:
    debug: true
    flow.mode: stream
```

This stream will process network packets as soon as they will be queued into `logisland_input_packets_topic` Kafka topic, each packet will be parsed as a record which will be pushed back to Kafka in the `logisland_parsed_packets_topic` topic.

Stream 2: Index the processed records into Elasticsearch

The second Kafka stream will handle `Records` pushed into the `logisland_parsed_packets_topic` topic to index them into ElasticSearch. So there is no need to define an output topic:

```
# Indexing
- stream: indexing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: processor
  documentation: a processor that pushes events to ES
  configuration:
    kafka.input.topics: logisland_parsed_packets_topic
    kafka.output.topics: none
    kafka.error.topics: logisland_error_packets_topic
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: none
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 2
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
```

The only processor in the processor chain of this stream is the `BulkAddElasticsearch` processor.

```
# Bulk add into ElasticSearch
- processor: ES Publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: A processor that pushes network packet records into ES
  configuration:
    elasticsearch.client.service: elasticsearch_service
    default.index: packets_index
    default.type: events
    timebased.index: today
    es.index.field: search_index
    es.type.field: record_type
```

The `default.index: packets_index` configuration parameter tells the elasticsearch processor to index records into an index starting with the `packets_index` string. The `timebased.index: today` configuration parameter tells the processor to use the current date after the index prefix. Thus the index name is of the form `/packets_index.2017.03.30`.

Finally, the `es.type.field: record_type` configuration parameter tells the processor to use the record field `record_type` of the incoming record to determine the ElasticSearch type to use within the index.

5. Stream network packets into the system

Let's install and use the Apache pycapa probe to capture and send packets to kafka topics in real time.

Install pycapa probe

All required steps to install pycapa probe are explained in this site, but here are the main installation steps :

1. Install libpcap, pip (python-pip) and python-devel :

```
yum install libpcap
yum install python-pip
yum install python-devel
```

2. Build pycapa probe from Metron repo

```
cd /tmp
git clone https://github.com/apache/incubator-metron.git
cd incubator-metron/metron-sensors/pycapa
pip install -r requirements.txt
python setup.py install
```

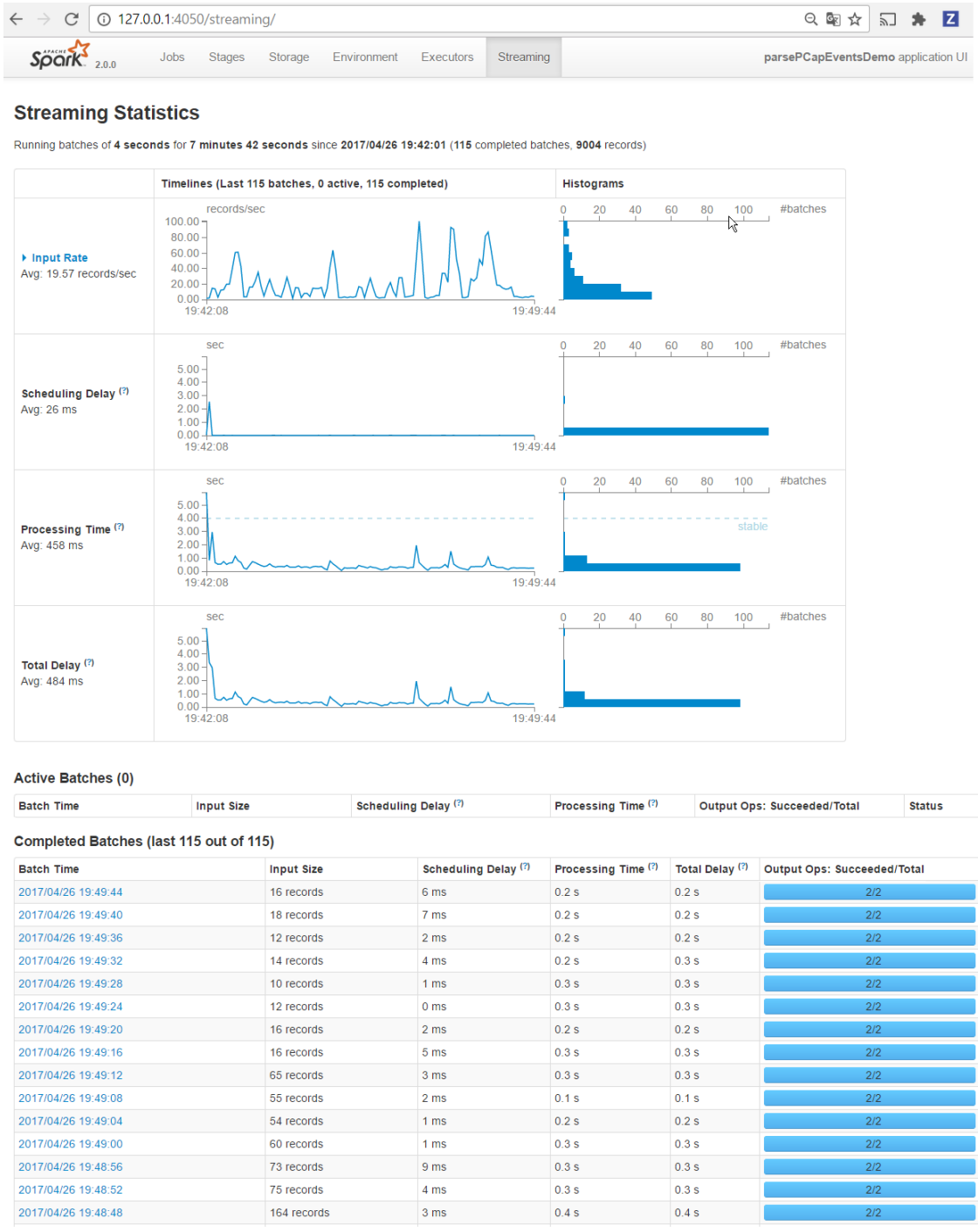
Capture network packets

To start capturing network packets into the topic `logisland_input_packets_topic` using pycapa probe, use the following command :

```
pycapa --producer --kafka sandbox:9092 --topic logisland_input_packets_topic -i eth0
```

6. Monitor your spark jobs and Kafka topics

Now go to http://sandbox:4050/streaming/ to see how fast Spark can process your data

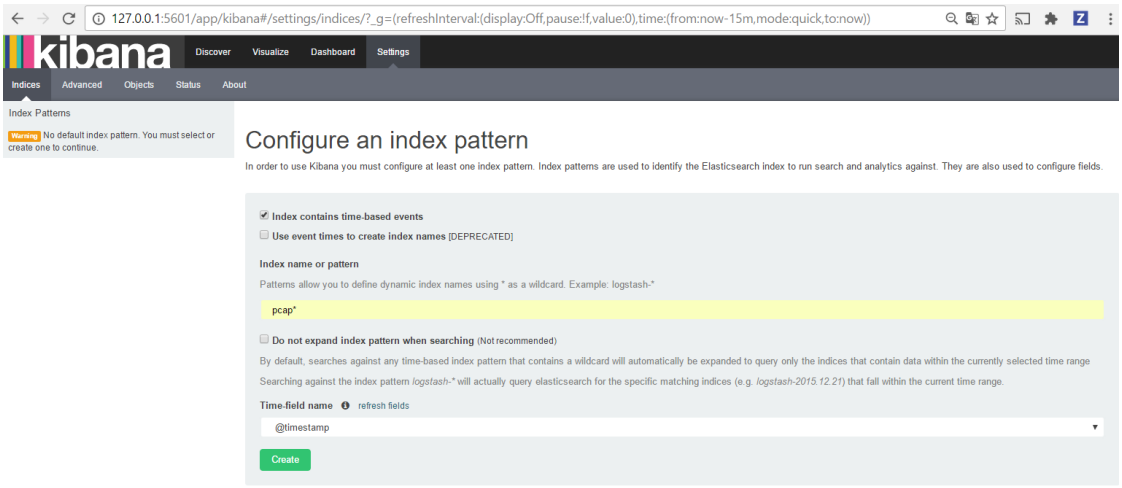


7. Use Kibana to inspect records

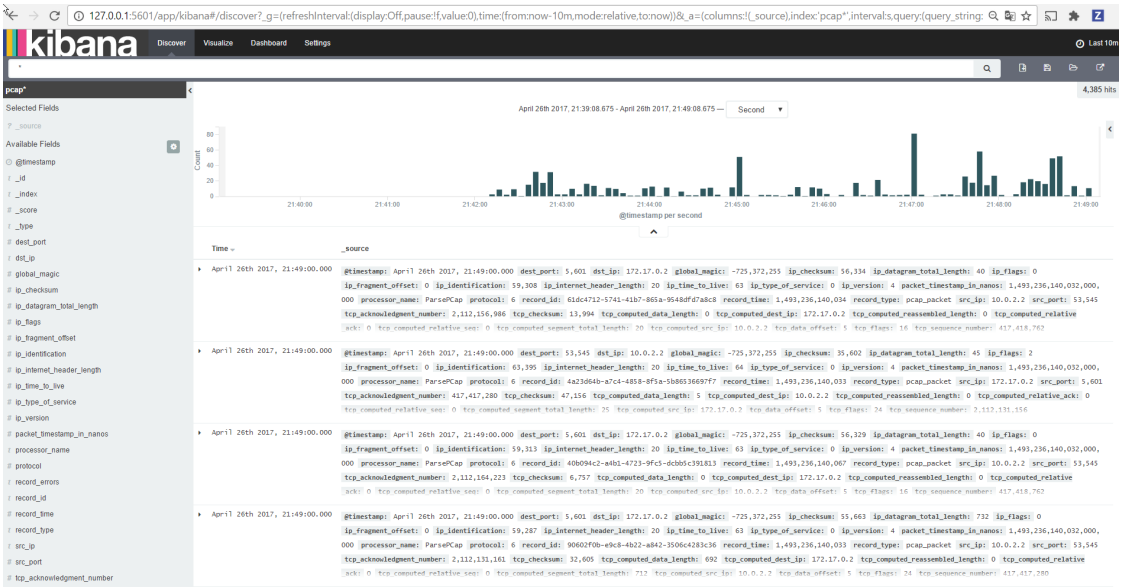
Inspect Network Packets under `Discover` tab

Open your browser and go to http://sandbox:5601/

Configure a new index pattern with `packets.*` as the pattern name and `@timestamp` as the time value field.



Then browse "Discover" tab, you should be able to explore your network packet records :



1.2.16 Generate Unique Ids

We will add a stage to the "index-apache-logs" tutorial. We will ensure every Record has a unique Id before injecting into Es. This way we are sure to not have documentAlreadyException or to have two records that overwrite themselves.

Note: If you are not familiar with logisland yet. You should really read "index-apache-logs" tutorial before this one.

We assume we are at the stage just before injecting apache logs into ES from "index-apache-logs"

Stream 1 : parse incoming apache log lines

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our ouput records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
# parsing
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that links
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    avro.output.schema: >
      {  "version":1,
         "type": "record",
         "name": "com.hurence.logisland.record.apache_log",
         "fields": [
           { "name": "record_errors",   "type": [ {"type": "array", "items": "string"}
↪,"null"] },
           { "name": "record_raw_key", "type": ["string","null"] },
           { "name": "record_raw_value", "type": ["string","null"] },
           { "name": "record_id",    "type": ["string"] },
           { "name": "record_time", "type": ["long"] },
           { "name": "record_type", "type": ["string"] },
           { "name": "src_ip",      "type": ["string","null"] },
           { "name": "http_method", "type": ["string","null"] },
           { "name": "bytes_out",   "type": ["long","null"] },
           { "name": "http_query",  "type": ["string","null"] },
           { "name": "http_version","type": ["string","null"] },
           { "name": "http_status", "type": ["string","null"] },
           { "name": "identd",      "type": ["string","null"] },
           { "name": "user",        "type": ["string","null"] }    ]}
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
  processorConfigurations:
```

Within this stream a `SplitText` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
# parse apache logs
- processor: apache_parser
  component: com.hurence.logisland.processor.SplitText
```

(continues on next page)

(continued from previous page)

```
  type: parser
  documentation: a parser that produce events from an apache log REGEX
  configuration:
    value.regex: (\S+)\s+(\S+)\s+(\S+)\s+\[([\w:\/]+\s[+\-]\d{4})\]\s+
↪"(\S+)\s+(\S+)\s*(\S*)"\s+(\S+)\s+(\S+)
    value.fields: src_ip,identd,user,record_time,http_method,http_query,http_version,
↪http_status,bytes_out
```

Within this stream a ModifyId processor takes Record ouput from SplitText processor and computes a new Id for them using the value of their field "record_raw_value" that should content the original line string of the apache log. It will hash it using "SHA-256" java implementation algorithm, using the charset "UTF-8".

parse apache logs - processor: apache_parser

component: com.hurence.logisland.processor.ModifyId type: parser documentation: a parser that modify record Ids configuration:

id.generation.strategy: hashFields hash.charset: UTF-8 fields.to.hash: record_raw_value hash.algorithm: SHA-256

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

Then you can process to your indexation in Elasticsearch as in "index-apache-logs" example.

1.2.17 Index JMS messages

In the following getting started tutorial, we'll explain you how to read messages from a JMS topic or queue and index them into an elasticsearch store.

The JMS data will leverage the JMS connector available as part of logisland connect.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

For kafka connect related information please follow as well the connectors section.

---

1. Installing ActiveMQ

In this tutorial we'll use Apache ActiveMQ.

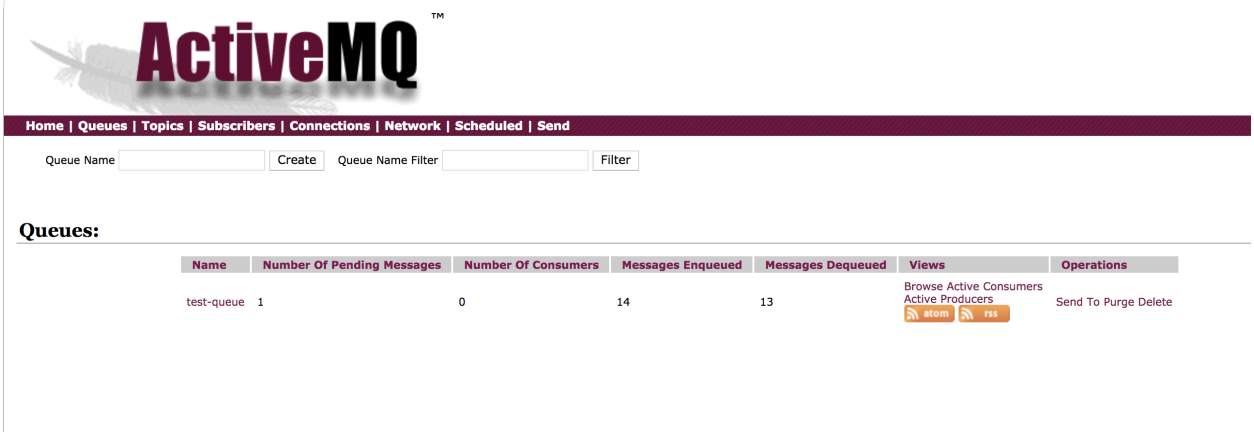
Once you downloaded the broker package just extract it in a folder and turn on your first broker by running:

```
bin/activemq start
```

You can verify if your broker is alive by connecting to the ActiveMQ console (login with admin/admin)

We are also going to create a test queue that we'll use for this tutorial.

To do that, in the just use the ActiveMQ console and in the *queue* section create a queue named *test-queue*. You should have your queue created as shown below.



As well, since JMS is actually an API, we have to provide to logisland the JMS connection factory and the client libraries. For this we can just copy the *activemq-all-5.15.5.jar* into the Logisland lib folder.

For instance, assuming you are running Logisland with the provided docker compose, you can just copy with a command like this:

..code-block:: bash

    docker cp ./activemq-all-5.15.5.jar logisland:/usr/local/logisland/lib

You can verify that activemq jar has been successfully copied inside the docker container by running

..code-block:: bash

    docker exec logisland ls lib/

2. Logisland job setup

For this tutorial please make sure to already have installed elasticsearch and JMS connector modules.

If not you can just do it through the componentes.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_5_4_0-
↪client:1.1.2

bin/components.sh -i com.datamountaineer:kafka-connect-jms:1.1.2
```

The interesting part in this tutorial is how to setup the JMS stream.

Let's first focus on the stream configuration and then on its pipeline in order to extract the data in the right way.

The JMS stream

Here we are going to use a special processor (`KafkaConnectStructuredSourceProviderService`) to use the kafka connect source as input for the structured stream defined below.

Logisland ships by default a kafka connect JMS source implemented by the class *com.datamountaineer.streamreactor.connect.jms.source.JMSSourceConnector*.

You can find more information about how to configure a JMS source in the official page of the JMS Connector

Coming back to our example, we would like to read from a queue called *test-queue* hosted in our local ActiveMQ broker. For this we will connect as usual to its Openwire channel and we'll use client acknowledgement to be sure to have an exactly once delivery.

The kafka connect controller service configuration will look like this:

```
- controllerService: kc_source_service
  component: com.hurence.logisland.stream.spark.provider.
↪KafkaConnectStructuredSourceProviderService
  configuration:
    kc.data.value.converter: com.hurence.logisland.connect.converter.
↪LogIslandRecordConverter
    kc.data.value.converter.properties: |
      record.serializer=com.hurence.logisland.serializer.KryoSerializer
    kc.data.key.converter.properties: |
      schemas.enable=false
    kc.data.key.converter: org.apache.kafka.connect.storage.StringConverter
    kc.worker.tasks.max: 1
    kc.connector.class: com.datamountaineer.streamreactor.connect.jms.source.
↪JMSSourceConnector
    kc.connector.offset.backing.store: memory
    kc.connector.properties: |
      connect.jms.url=tcp://sandbox:61616
      connect.jms.initial.context.factory=org.apache.activemq.jndi.
↪ActiveMQInitialContextFactory
      connect.jms.connection.factory=ConnectionFactory
      connect.jms.kcql=INSERT INTO topic SELECT * FROM test-queue WITHTYPE QUEUE
      connect.progress.enabled=true
```

The pipeline

Within this stream, a we need to extract the data coming from the JMS.

First of all a `FlatMap` processor takes out the value and key (required when using *StructuredStream* as source of records)

```
processorConfigurations:
 - processor: flatten
   component: com.hurence.logisland.processor.FlatMap
   type: processor
   documentation: "Takes out data from record_value"
   configuration:
     keep.root.record: false
```

Then, since our JMS messages will carry text data, we need to extract this data from the raw message bytes:

```
- processor: add_fields
  component: com.hurence.logisland.processor.AddFields
  type: processor
  documentation: "Extract the message as a text"
  configuration:
  conflict.resolution.policy: overwrite_existing
  message_text: ${new String(bytes_payload)}
```

Now we will as well set the record time as the time when the message has been created (and not received). This thanks to a NormalizeFields processor:

```
- processor: rename_fields
  component: com.hurence.logisland.processor.NormalizeFields
  type: processor
  documentation: "Change the record time according to message_timestamp field"
  configuration:
  conflict.resolution.policy: overwrite_existing
  record_time: message_timestamp
```

Last but not least, a `BulkAddElasticsearch` takes care of indexing a `Record` sending it to elasticsearch.

```
-  processor: es_publisher
   component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
   type: processor
   documentation: a processor that indexes processed events in elasticsearch
   configuration:
     elasticsearch.client.service: elasticsearch_service
     default.index: logisland
     default.type: event
     timebased.index: yesterday
     es.index.field: search_index
     es.type.field: record_type
```

In details, this processor makes use of a `Elasticsearch_5_4_0_ClientService` controller service to interact with our Elasticsearch 5.X backend running locally (and started as part of the docker compose configuration we mentioned above).

Here below its configuration:

```
- controllerService: elasticsearch_service
  component: com.hurence.logisland.service.elasticsearch.Elasticsearch_5_4_0_
↪ClientService
  type: service
  documentation: elasticsearch service
  configuration:
    hosts: sandbox:9300
    cluster.name: es-logisland
    batch.size: 5000
```

3. Launch the script

Connect a shell to your logisland container to launch the following streaming jobs.

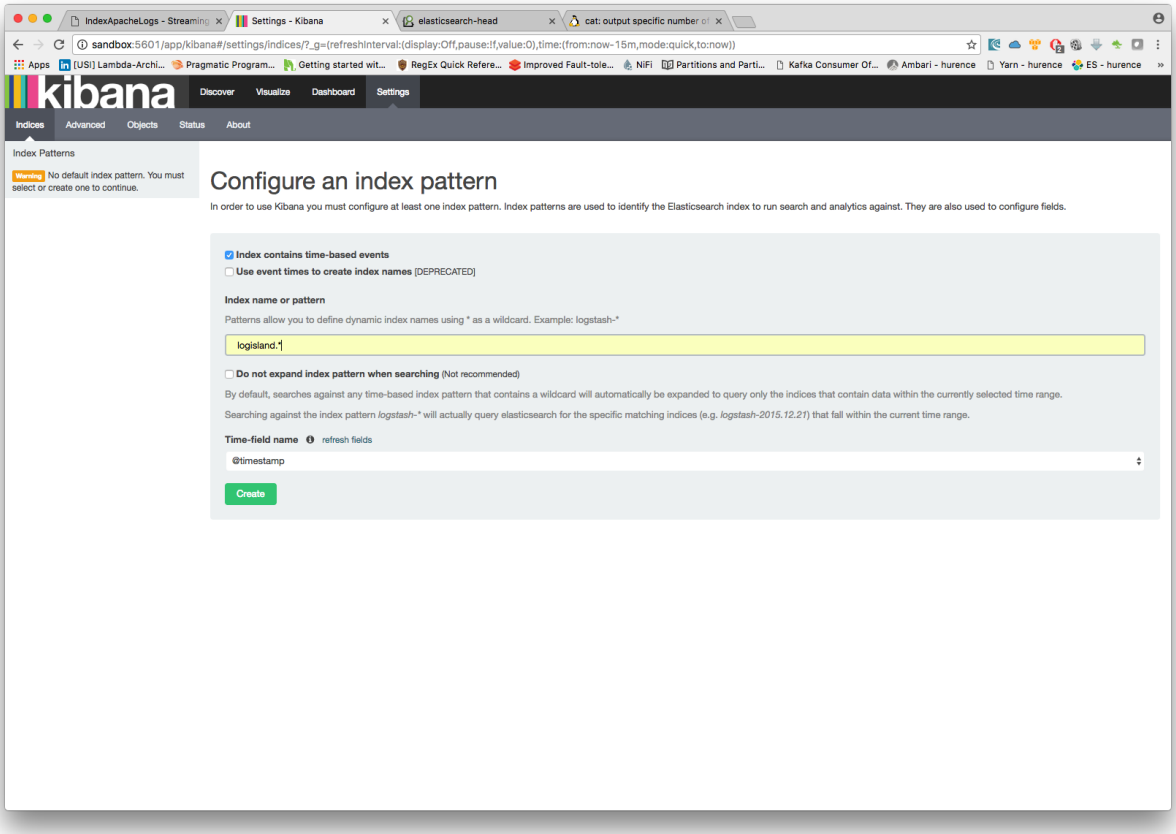
```
bin/logisland.sh --conf conf/index-jms-messages.yml
```

4. Do some insights and visualizations

With ElasticSearch, you can use Kibana.

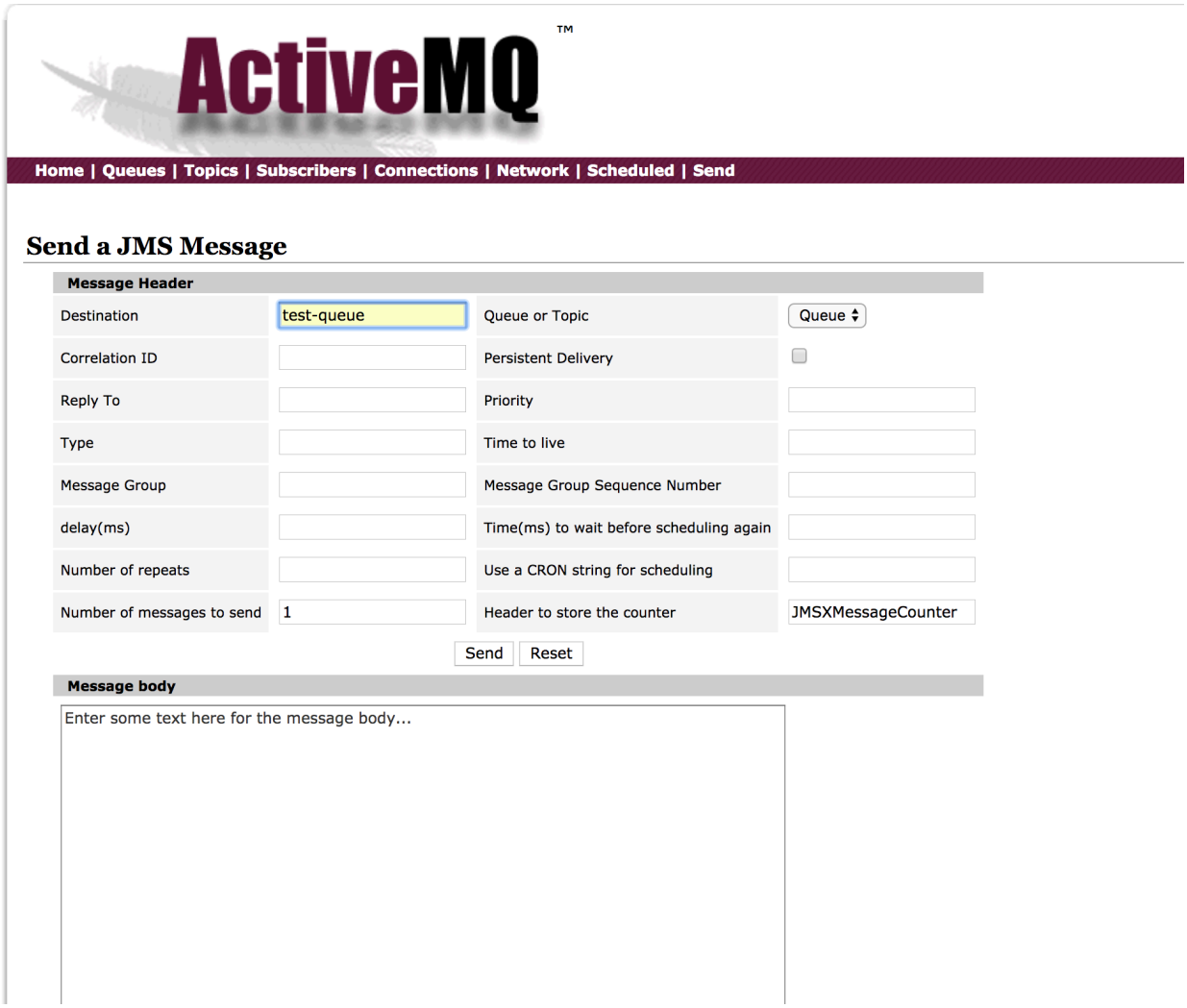
Open up your browser and go to http://sandbox:5601/app/kibana#/ and you should be able to explore the blockchain transactions.

Configure a new index pattern with `logisland.*` as the pattern name and `@timestamp` as the time value field.

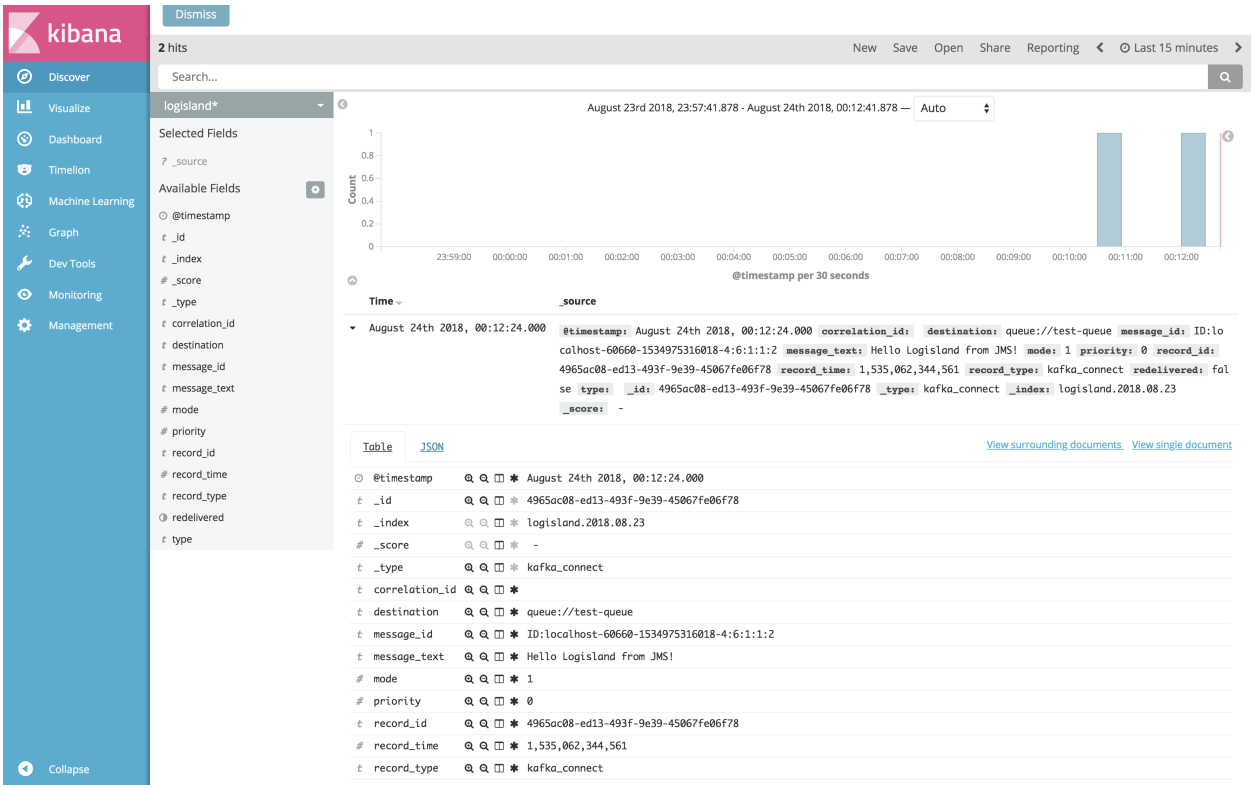


Now just send some message thanks to the ActiveMQ console.

Click on the *Send* link on the top of the console main page and specify the destination to *test-queue* and type the message you like. You should have something like this:

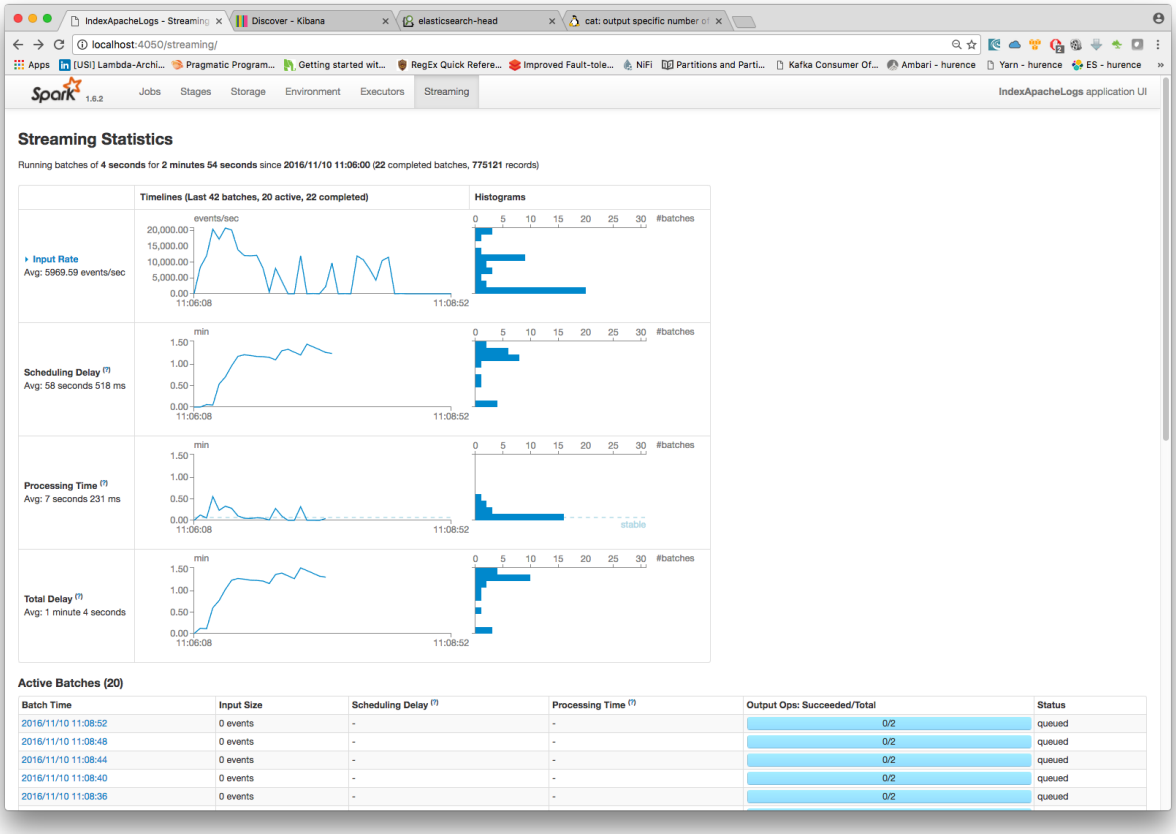


Now that the message have been consumed (you can also verify this thanks to the ActiveMQ console) you can come back to kibana and go to Explore panel for the latest 15' time window you'll only see logisland process_metrics events which give you insights about the processing bandwidth of your streams.

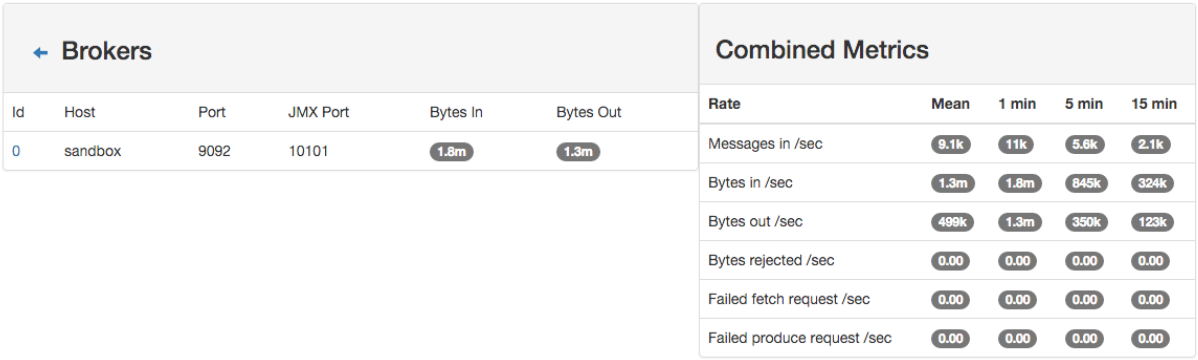


5. Monitor your spark jobs and Kafka topics

Now go to http://sandbox:4050/streaming/ to see how fast Spark can process your data



Another tool can help you to tweak and monitor your processing http://sandbox:9000/



1.2.18 Index blockchain transactions

In the following getting started tutorial, we'll explain you how to leverage logisland connectors flexibility in order process in real time every transaction emitted by the bitcoin blockchain platform and index each record into an elasticsearch platform.

This will allow us to run some dashboarding and visual data analysis as well.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

For kafka connect related information please follow as well the connectors section.

1. Logisland job setup

Install the blockchain connector if not already done.

```
bin/components.sh -i com.datamountaineer:kafka-connect-blockchain:1.1.2
```

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here for ElasticSearch :

```
vim conf/index-blockchain-transactions.yml
```

We will start by explaining each part of the config file.

The engine

The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode.

```
engine:
   component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
   type: engine
   documentation: Index some blockchain transactions with logisland
   configuration:
     spark.app.name: BlockchainTest
     spark.master: local[*]
     spark.driver.memory: 512M
     spark.driver.cores: 1
     spark.executor.memory: 512M
     spark.executor.instances: 4
     spark.executor.cores: 2
     spark.yarn.queue: default
     spark.yarn.maxAppAttempts: 4
     spark.yarn.am.attemptFailuresValidityInterval: 1h
     spark.yarn.max.executor.failures: 20
     spark.yarn.executor.failuresValidityInterval: 1h
     spark.task.maxFailures: 8
     spark.serializer: org.apache.spark.serializer.KryoSerializer
     spark.streaming.batchDuration: 2000
     spark.streaming.backpressure.enabled: false
     spark.streaming.blockInterval: 500
     spark.streaming.kafka.maxRatePerPartition: 10000
     spark.streaming.timeout: -1
     spark.streaming.unpersist: false
     spark.streaming.kafka.maxRetries: 3
     spark.streaming.ui.retainedBatches: 200
     spark.streaming.receiver.writeAheadLog.enable: false
     spark.ui.port: 4040

 The `controllerServiceConfigurations` part is here to define all services that be␣
↪shared by processors within the whole job.

 ==================
 The parsing stream
```

(continues on next page)

(continued from previous page)

```
==================

Here we are going to use a special processor␣
↪(``KafkaConnectStructuredSourceProviderService``) to use the kafka connect source␣
↪as input for the structured stream defined below.

For this example, we are going to use the source *com.datamountaineer.streamreactor.
↪connect.blockchain.source.BlockchainSourceConnector*
that opens a secure websocket connections to the blockchain subscribing to any␣
↪transaction update stream.


.. code-block:: yaml

    ControllerServiceConfigurations:
    - controllerService: kc_source_service
      component: com.hurence.logisland.stream.spark.provider.
↪KafkaConnectStructuredSourceProviderService
      configuration:
        kc.data.value.converter: com.hurence.logisland.connect.converter.
↪LogIslandRecordConverter
        kc.data.value.converter.properties: |
          record.serializer=com.hurence.logisland.serializer.KryoSerializer
        kc.data.key.converter.properties: |
          schemas.enable=false
        kc.data.key.converter: org.apache.kafka.connect.storage.StringConverter
        kc.worker.tasks.max: 1
        kc.connector.class: com.datamountaineer.streamreactor.connect.blockchain.
↪source.BlockchainSourceConnector
        kc.connector.offset.backing.store: memory
        kc.connector.properties: |
          connect.blockchain.source.url=wss://ws.blockchain.info/inv
          connect.blockchain.source.kafka.topic=blockchain
```

Note: Our source is providing structured value hence we convert with LogInslandRecordConverter serializing with Kryo

```
# Kafka sink configuration
- controllerService: kafka_out_service
  component: com.hurence.logisland.stream.spark.structured.provider.
↪KafkaStructuredStreamProviderService
  configuration:
    kafka.output.topics: logisland_raw
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

So that, we can now define the *parsing stream* using those source and sink

```
######### parsing stream #############
- stream: parsing_stream_source
  component: com.hurence.logisland.stream.spark.structured.StructuredStream
  documentation: "Takes records from the kafka source and distributes related␣
↪partitions over a kafka topic. Records are then handed off to the indexing stream"
  configuration:
    read.topics: /a/in
    read.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.stream.service.provider: kc_source_service
    write.topics: logisland_raw
    write.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.stream.service.provider: kafka_out_service
```

Within this stream, a `FlatMap` processor takes out the value and key (required when using *StructuredStream* as source of records)

```
processorConfigurations:
 - processor: flatten
   component: com.hurence.logisland.processor.FlatMap
   type: processor
   documentation: "Takes out data from record_value"
   configuration:
     keep.root.record: false
     copy.root.record.fields: true
```

The indexing stream

Inside this engine, you will run a Kafka stream of processing, so we set up input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our output records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
- stream: parsing_stream_source
  component: com.hurence.logisland.stream.spark.structured.StructuredStream
  documentation: "Takes records from the kafka source and distributes related␣
↪partitions over a kafka topic. Records are then handed off to the indexing stream"
  configuration:
    read.topics: /a/in
    read.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.stream.service.provider: kc_source_service
    write.topics: logisland_raw
    write.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.stream.service.provider: kafka_out_service
```

Within this stream, a `BulkAddElasticsearch` takes care of indexing a `Record` sending it to elasticsearch.

```
- processor: es_publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: a processor that indexes processed events in elasticsearch
  configuration:
    elasticsearch.client.service: elasticsearch_service
    default.index: logisland
    default.type: event
    timebased.index: yesterday
    es.index.field: search_index
    es.type.field: record_type
```

In details, this processor makes use of a `Elasticsearch_5_4_0_ClientService` controller service to interact with our Elasticsearch 5.X backend running locally (and started as part of the docker compose configuration we mentioned above).

Here below its configuration:

```
- controllerService: elasticsearch_service
  component: com.hurence.logisland.service.elasticsearch.Elasticsearch_5_4_0_
↪ClientService
  type: service
  documentation: elasticsearch service
  configuration:
    hosts: sandbox:9300
    cluster.name: es-logisland
    batch.size: 5000
```

2. Launch the script

Connect a shell to your logisland container to launch the following streaming jobs.

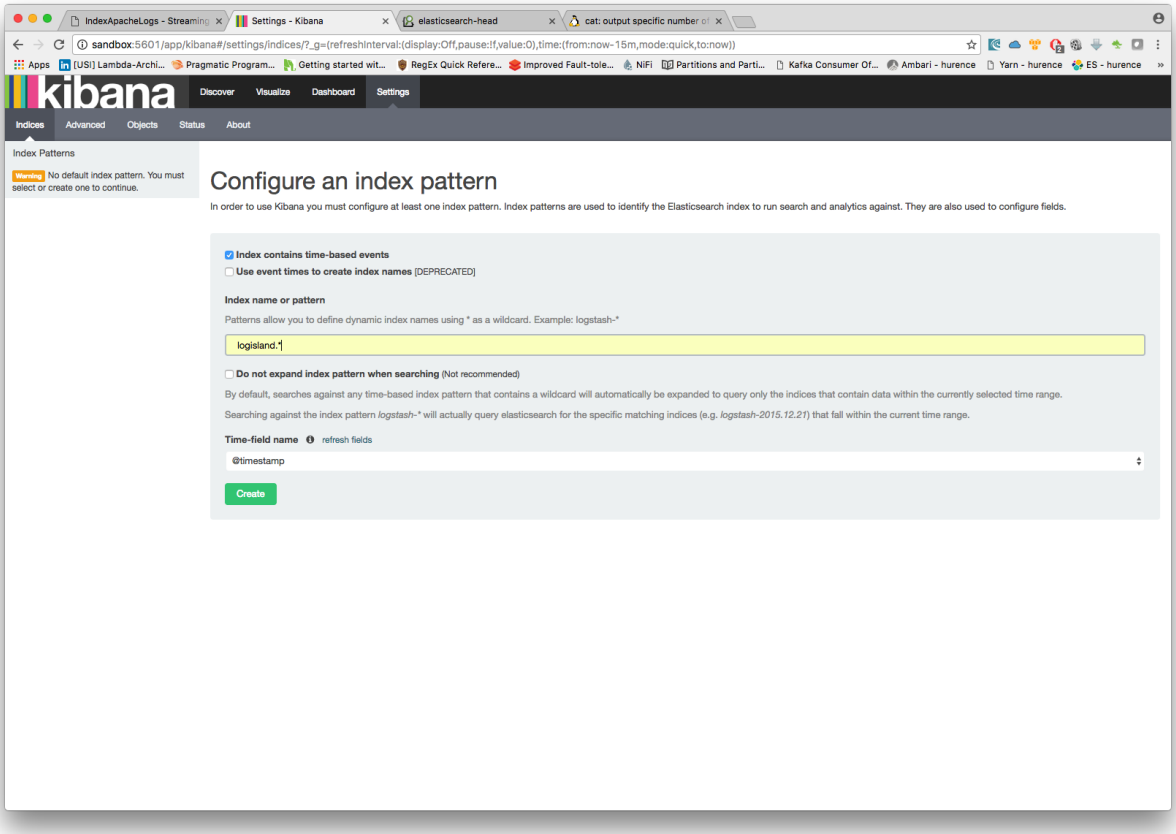
```
bin/logisland.sh --conf conf/index-blockchain-transactions.yml
```

3. Do some insights and visualizations

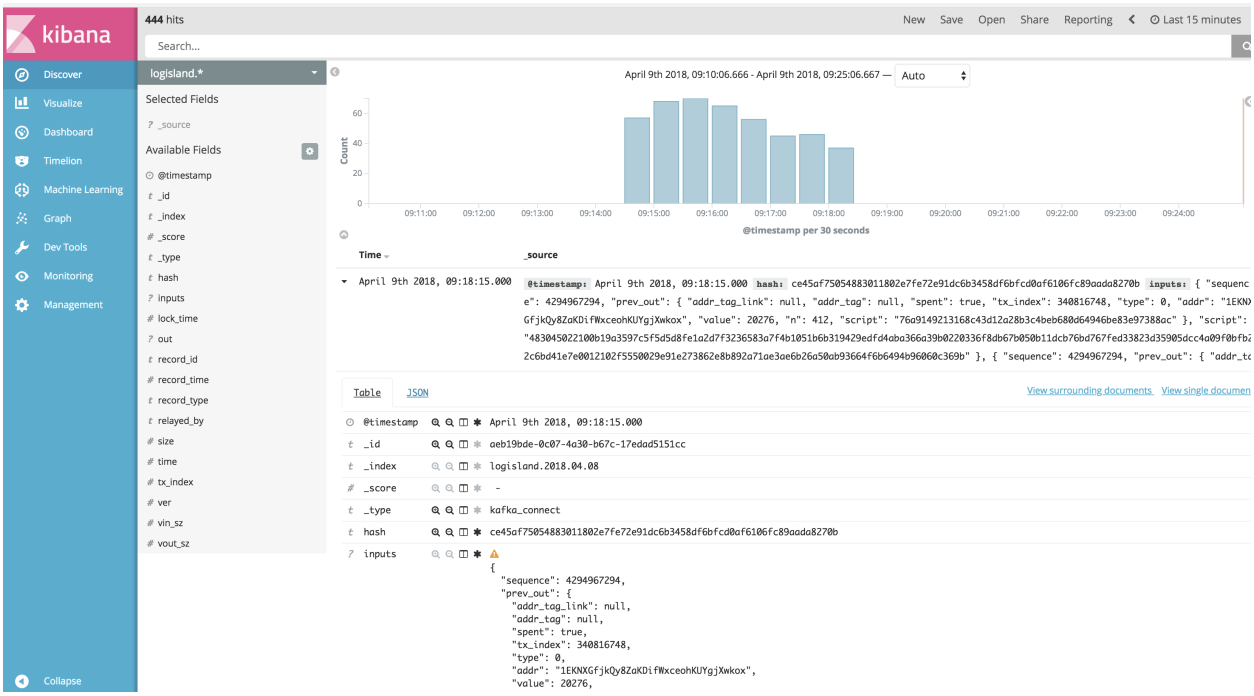
With ElasticSearch, you can use Kibana.

Open up your browser and go to http://sandbox:5601/app/kibana#/ and you should be able to explore the blockchain transactions.

Configure a new index pattern with `logisland.*` as the pattern name and `@timestamp` as the time value field.



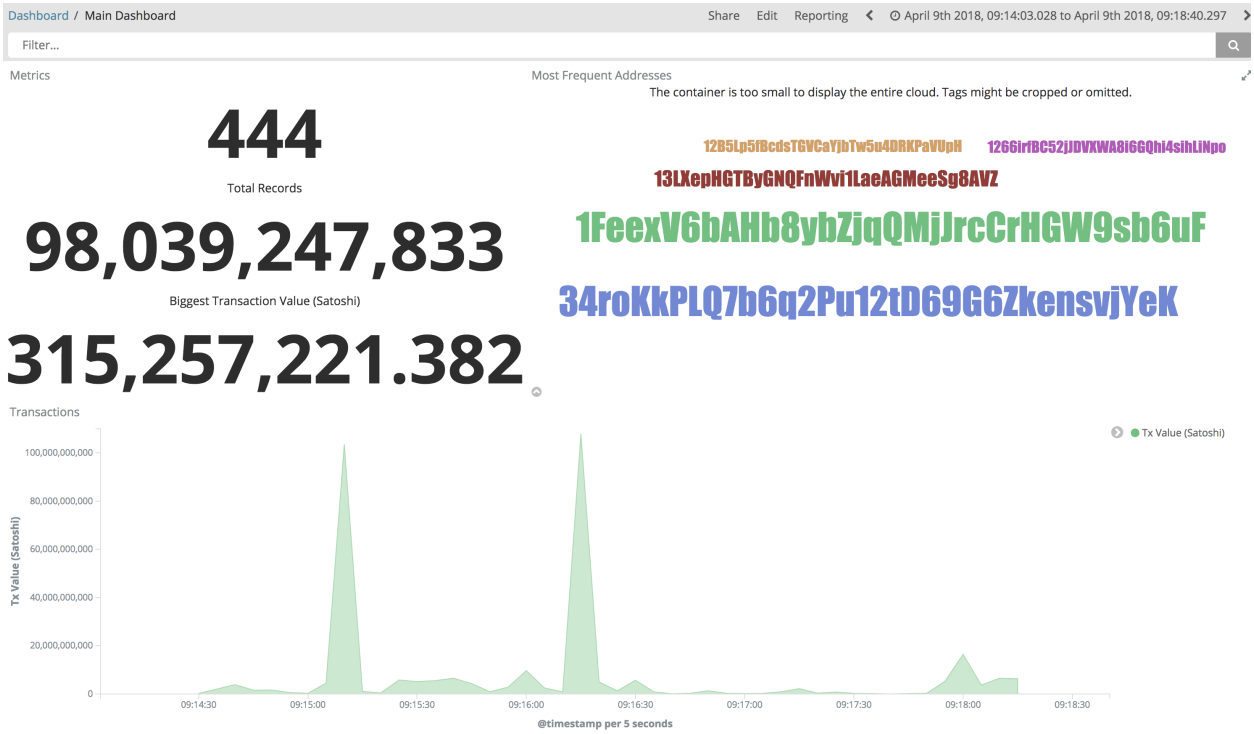
Then if you go to Explore panel for the latest 15' time window you'll only see logisland process_metrics events which give you insights about the processing bandwidth of your streams.



You can try as well to create some basic visualization in order to draw the total satoshi transacted amount (aggregating

sums of `out.value` field).

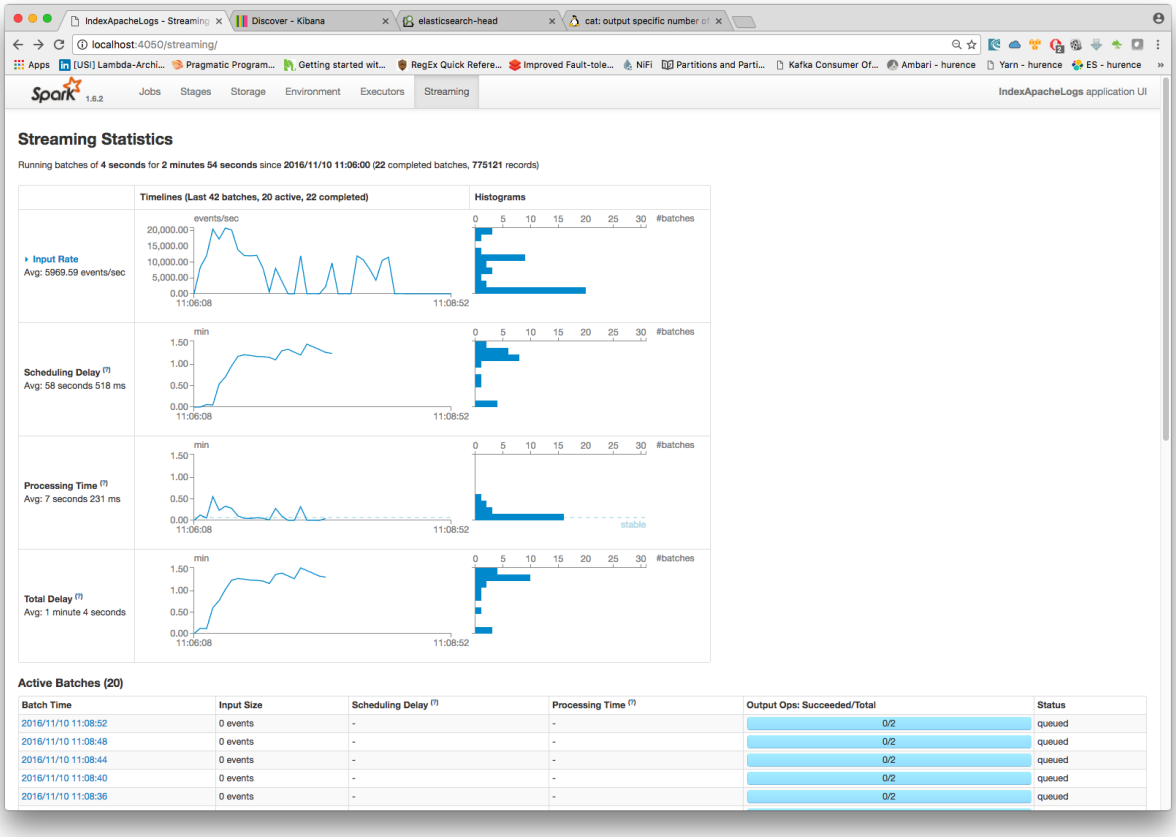
Below a nice example:



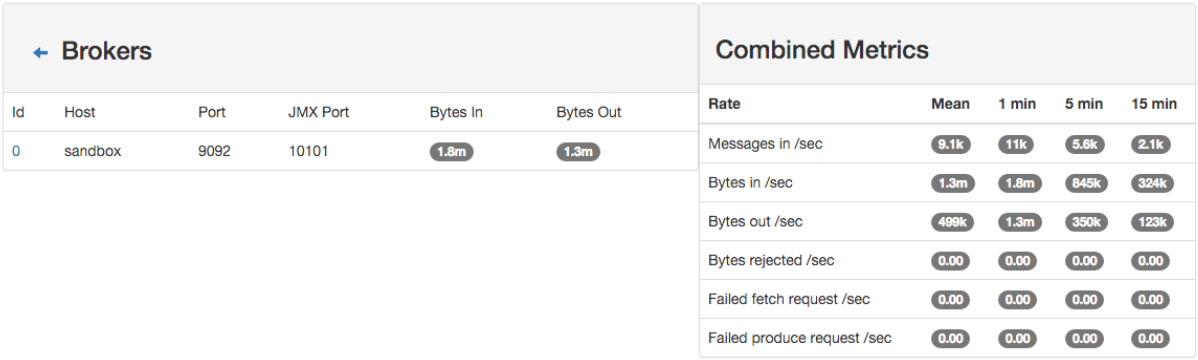
Ready to discover which addresses received most of the money? Give it a try ;-)

4. Monitor your spark jobs and Kafka topics

Now go to http://sandbox:4050/streaming/ to see how fast Spark can process your data



Another tool can help you to tweak and monitor your processing http://sandbox:9000/



1.2.19 Extract Records from Excel File

In the following getting started tutorial we'll drive you through the process of extracting data from any Excel file with LogIsland platform.

Both XLSX and old XLS file format are supported.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

---

Note, it is possible to store data in different datastores. In this tutorial, we will see the case of ElasticSearch only.

1.Install required components

For this tutorial please make sure to already have installed elasticsearch and excel modules. If not you can just do it through the componentes.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_5_4_0-
↪client:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-processor-excel:1.1.2
```

2. Logisland job setup

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here for ElasticSearch :

```
docker exec -i -t logisland vim conf/index-excel-spreadsheet.yml
```

We will start by explaining each part of the config file.

An Engine is needed to handle the stream processing. This `conf/extract-excel-data.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode with 2 cpu cores and 2G of RAM.

```
  engine:
component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
type: engine
documentation: Index records of an excel file with LogIsland
configuration:
  spark.app.name: IndexExcelDemo
  spark.master: local[4]
  spark.driver.memory: 1G
  spark.driver.cores: 1
  spark.executor.memory: 2G
  spark.executor.instances: 4
  spark.executor.cores: 2
  spark.yarn.queue: default
  spark.yarn.maxAppAttempts: 4
  spark.yarn.am.attemptFailuresValidityInterval: 1h
  spark.yarn.max.executor.failures: 20
  spark.yarn.executor.failuresValidityInterval: 1h
  spark.task.maxFailures: 8
  spark.serializer: org.apache.spark.serializer.KryoSerializer
  spark.streaming.batchDuration: 1000
  spark.streaming.backpressure.enabled: false
  spark.streaming.unpersist: false
  spark.streaming.blockInterval: 500
  spark.streaming.kafka.maxRatePerPartition: 3000
  spark.streaming.timeout: -1
  spark.streaming.unpersist: false
  spark.streaming.kafka.maxRetries: 3
  spark.streaming.ui.retainedBatches: 200
```

(continues on next page)

(continued from previous page)

```
  spark.streaming.receiver.writeAheadLog.enable: false
  spark.ui.port: 4050
```

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job, here an Elasticsearch service that will be used later in the `BulkAddElasticsearch` processor.

```
- controllerService: elasticsearch_service
  component: com.hurence.logisland.service.elasticsearch.Elasticsearch_5_4_0_
↪ClientService
  type: service
  documentation: elasticsearch service
  configuration:
    hosts: sandbox:9300
    cluster.name: es-logisland
    batch.size: 5000
```

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

We can define some serializers to marshall all records from and to a topic. We assume that the stream will be serializing the input file as a byte array in a single record. Reason why we will use a ByteArraySerialiser in the configuration below.

```
# main processing stream
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that converts raw excel file content into structured log␣
↪records
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.
↪BytesArraySerializer
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

Within this stream, an `ExcelExtract` processor takes a byte array excel file content and computes a list of `Record`.

```
# parse excel cells into records
    - processor: excel_parser
      component: com.hurence.logisland.processor.excel.ExcelExtract
      type: parser
      documentation: a parser that produce events from an excel file
      configuration:
        record.type: excel_record
        skip.rows: 1
        field.names: segment,country,product,discount_band,units_sold,manufacturing,
↪sale_price,gross_sales,discounts,sales,cogs,profit,record_time,month_number,month_
↪name,year
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

Note: Please note that we are mapping the excel column *Date* to be the timestamp of the produced record (*record_time* field) in order to use this as time reference in elasticsearch/kibana (see below).

The second processor will handle `Records` produced by the `ExcelExtract` to index them into elasticsearch

```
# add to elasticsearch
- processor: es_publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: a processor that trace the processed events
  configuration:
    elasticsearch.client.service: elasticsearch_service
    default.index: logisland
    default.type: event
    timebased.index: yesterday
    es.index.field: search_index
    es.type.field: record_type
```

3. Launch the script

For this tutorial we will handle an excel file. We will process it with an ExcelExtract that will produce a bunch of Records and we'll send them to Elastiscearch Connect a shell to your logisland container to launch the following streaming jobs.

For ElasticSearch :

```
docker exec -i -t logisland bin/logisland.sh --conf conf/index-excel-spreadsheet.yml
```

4. Inject an excel file into the system

Now we're going to send a file to `logisland_raw` Kafka topic.

For testing purposes, we will use kafkacat, a *generic command line non-JVM Apache Kafka producer and consumer* which can be easily installed.

Note: Sending raw files through kafka is not recommended for production use since kafka is designed for high throughput and not big message size.

The configuration above is suited to work with the example file *Financial Sample.xlsx*.

Let's send this file in a single message to LogIsland with kafkacat to `logisland_raw` Kafka topic

```
kafkacat -P -t logisland_raw -v -b sandbox:9092 ./Financial\ Sample.xlsx
```

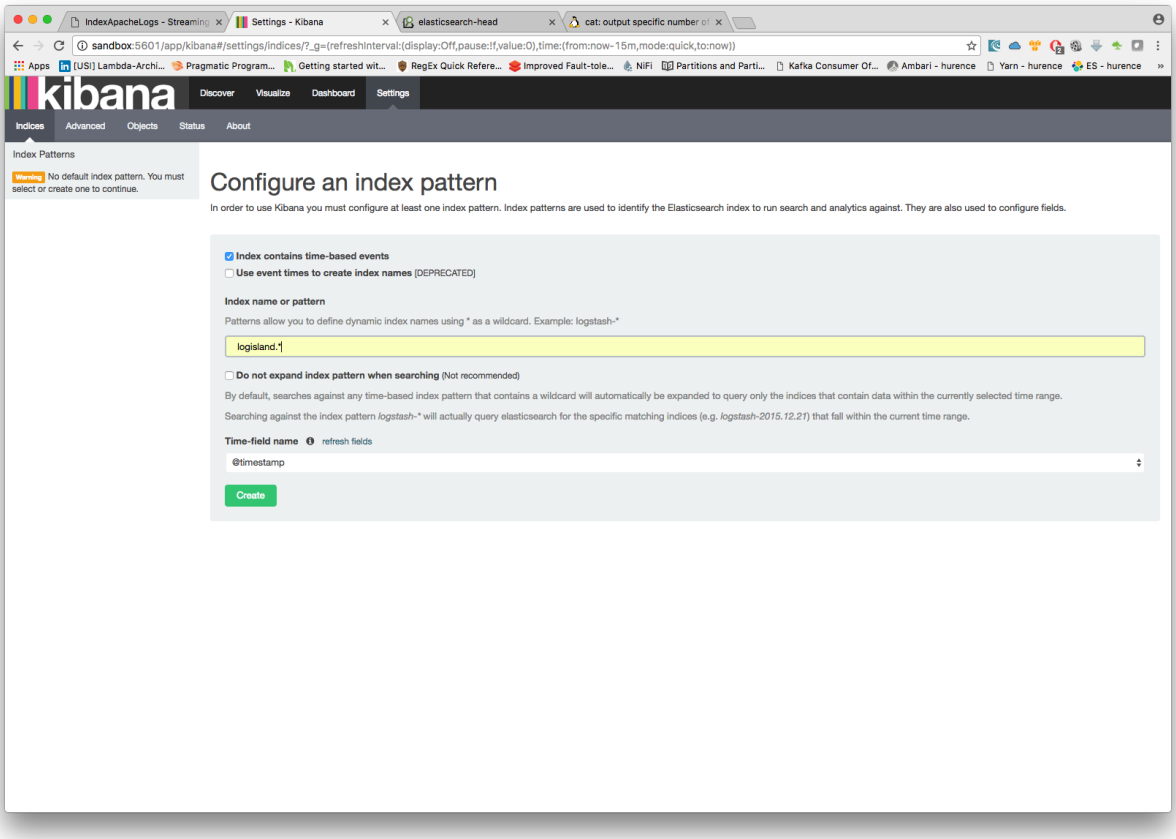
5. Inspect the logs

Kibana

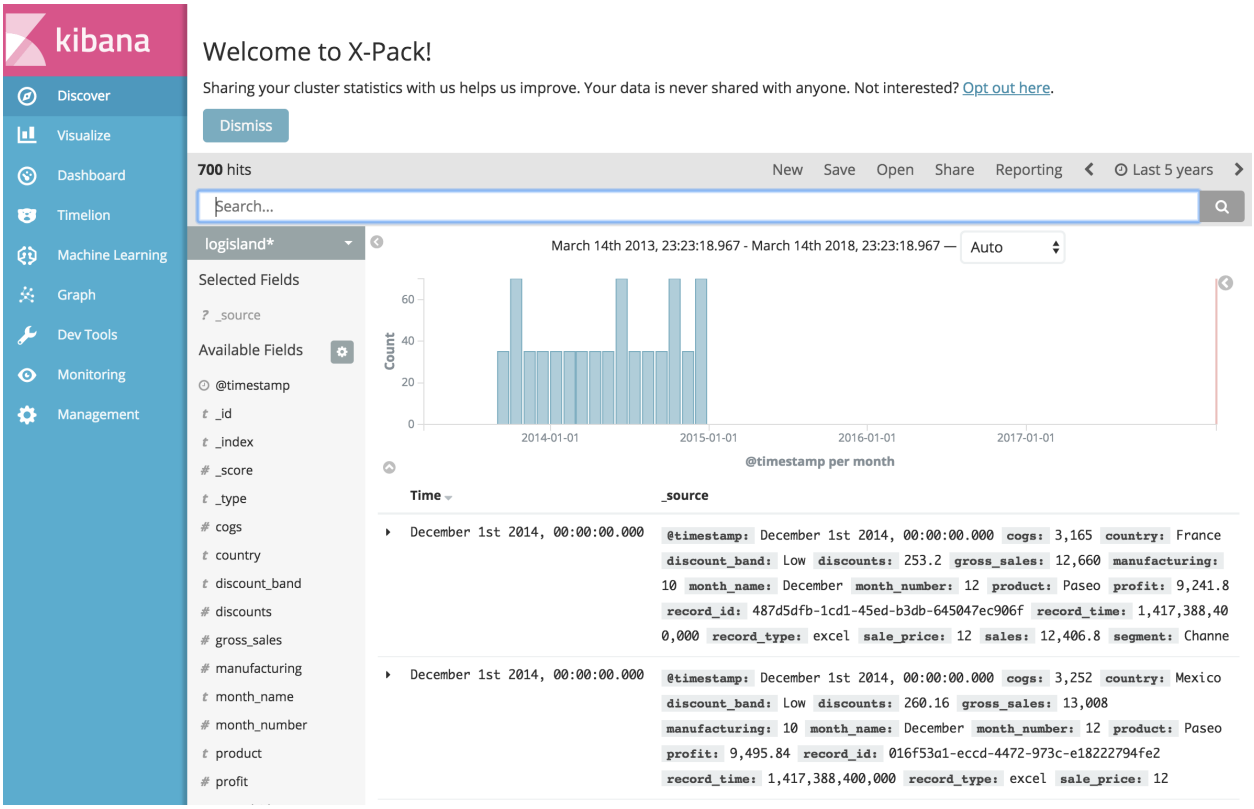
With ElasticSearch, you can use Kibana.

Open up your browser and go to http://sandbox:5601/ and you should be able to explore your excel records.

Configure a new index pattern with `logisland.*` as the pattern name and `@timestamp` as the time value field.



Then if you go to Explore panel for the latest 5 years time window. You are now able to play with the indexed data.



Thanks logisland! :-)

1.2.20 IIoT with MQTT and Logisland Data-Historian

In the following getting tutorial we'll drive you through the process of IIoT enablement with LogIsland platform.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

---

docker run -td –name kapua-sql -p 8181:8181 -p 3306:3306 kapua/kapua-sql:0.3.2 docker run -td –name kapua-elasticsearch -p 9200:9200 -p 9300:9300 elasticsearch:5.4.0 -Ecluster.name=kapua-datastore -Ediscovery.type=single-node -Etransport.host=_site_ -Etransport.ping_schedule=-1 -Etransport.tcp.connect_timeout=30s docker run -td –name kapua-broker –link kapua-sql:db –link kapua-elasticsearch:es –env commons.db.schema.update=true -p 1883:1883 -p 61614:61614 kapua/kapua-broker:0.3.2 docker run -td –name kapua-console –link kapua-sql:db –link kapua-broker:broker –link kapua-elasticsearch:es –env commons.db.schema.update=true -p 8080:8080 kapua/kapua-console:0.3.2 docker run -td –name kapua-api –link kapua-sql:db –link kapua-broker:broker –link kapua-elasticsearch:es –env commons.db.schema.update=true -p 8081:8080 kapua/kapua-api:0.3.2

docker run -td –name logisland-historian -p 8983:8983 hurence/chronix:latest

docker run -it –env MQTT_BROKER_URL=tcp://10.20.20.87:1883 –env SOLR_CONNECTION=http://10.20.20.87:8983/solr –name kapua-logisland hurence/logisland:0.12.0 bin/logisland.sh –conf conf/mqtt-to-historian.yml

Note, it is possible to store data in different datastores. In this tutorial, we will see the case of ElasticSearch and Solr.

1. Logisland job setup

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here for ElasticSearch :

```
docker exec -i -t logisland vim conf/index-apache-logs.yml
```

And here for Solr :

```
docker exec -i -t logisland vim conf/index-apache-logs-solr.yml
```

We will start by explaining each part of the config file.

An Engine is needed to handle the stream processing. This `conf/index-apache-logs.yml` configuration file defines a stream processing job setup. The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode with 2 cpu cores and 2G of RAM.

```
engine:
  component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
  type: engine
  documentation: Index some apache logs with logisland
  configuration:
    spark.app.name: IndexApacheLogsDemo
    spark.master: local[2]
    spark.driver.memory: 1G
    spark.driver.cores: 1
    spark.executor.memory: 2G
    spark.executor.instances: 4
    spark.executor.cores: 2
    spark.yarn.queue: default
    spark.yarn.maxAppAttempts: 4
    spark.yarn.am.attemptFailuresValidityInterval: 1h
    spark.yarn.max.executor.failures: 20
    spark.yarn.executor.failuresValidityInterval: 1h
    spark.task.maxFailures: 8
    spark.serializer: org.apache.spark.serializer.KryoSerializer
    spark.streaming.batchDuration: 1000
    spark.streaming.backpressure.enabled: false
    spark.streaming.unpersist: false
    spark.streaming.blockInterval: 500
    spark.streaming.kafka.maxRatePerPartition: 3000
    spark.streaming.timeout: -1
    spark.streaming.unpersist: false
    spark.streaming.kafka.maxRetries: 3
    spark.streaming.ui.retainedBatches: 200
    spark.streaming.receiver.writeAheadLog.enable: false
    spark.ui.port: 4050
```

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job, here an Elasticsearch service that will be used later in the `BulkAddElasticsearch` processor.

```
- controllerService: elasticsearch_service
  component: com.hurence.logisland.service.elasticsearch.Elasticsearch_5_4_0_
↪ClientService
  type: service
  documentation: elasticsearch service
  configuration:
    hosts: sandbox:9300
```

(continues on next page)

(continued from previous page)

```
    cluster.name: es-logisland
    batch.size: 5000
```

Inside this engine you will run a Kafka stream of processing, so we setup input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our ouput records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
- stream: parsing_stream
  component: com.hurence.logisland.stream.spark.KafkaRecordStreamParallelProcessing
  type: stream
  documentation: a processor that converts raw apache logs into structured log records
  configuration:
    kafka.input.topics: logisland_raw
    kafka.output.topics: logisland_events
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: none
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

Within this stream a `SplitText` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
# parse apache logs
- processor: apache_parser
  component: com.hurence.logisland.processor.SplitText
  type: parser
  documentation: a parser that produce events from an apache log REGEX
  configuration:
    value.regex: (\S+)\s+(\S+)\s+(\S+)\s+\[([\w:\/]+\s[+\-]\d{4})\]\s+
↪"(\S+)\s+(\S+)\s*(\S*)"\s+(\S+)\s+(\S+)
    value.fields: src_ip,identd,user,record_time,http_method,http_query,http_version,
↪http_status,bytes_out
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be parsed as an event which will be pushed back to Kafka in the `logisland_events` topic.

The second processor will handle `Records` produced by the `SplitText` to index them into elasticsearch

```
# add to elasticsearch
- processor: es_publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: a processor that trace the processed events
  configuration:
    elasticsearch.client.service: elasticsearch_service
```

(continues on next page)

(continued from previous page)

```
    default.index: logisland
    default.type: event
    timebased.index: yesterday
    es.index.field: search_index
    es.type.field: record_type
```

Solr

In the case of Solr, we have to declare another service :

```
# Datastore service using Solr 6.6.2 - 5.5.5 also available
- controllerService: datastore_service
  component: com.hurence.logisland.service.solr.Solr_6_6_2_ClientService
  type: service
  documentation: "SolR 6.6.2 service"
  configuration:
    solr.cloud: false
    solr.connection.string: http://sandbox:8983/solr
    solr.collection: solr-apache-logs
    solr.concurrent.requests: 4
    flush.interval: 2000
    batch.size: 1000
```

With this configuration, Solr is used in standalone mode but you can also use the cloud mode by changing the corresponding config.

Note: You have to create the core/collection manually with the following fields : `src_ip`, `identd`, `user`, `bytes_out`, `http_method`, `http_version`, `http_query`, `http_status`

Then, the second processor have to send data to Solr :

```
# all the parsed records are added to solr by bulk
- processor: solr_publisher
  component: com.hurence.logisland.processor.datastore.BulkPut
  type: processor
  documentation: "indexes processed events in SolR"
  configuration:
    datastore.client.service: datastore_service
```

2. Launch the script

For this tutorial we will handle some apache logs with a splitText parser and send them to Elastiscearch Connect a shell to your logisland container to launch the following streaming jobs.

For ElasticSearch :

```
docker exec -i -t logisland bin/logisland.sh --conf conf/index-apache-logs.yml
```

For Solr :

```
docker exec -i -t logisland bin/logisland.sh --conf conf/index-apache-logs-solr.yml
```

3. Inject some Apache logs into the system

Now we're going to send some logs to `logisland_raw` Kafka topic.

We could setup a logstash or flume agent to load some apache logs into a kafka topic but there's a super useful tool in the Kafka ecosystem : kafkacat, a *generic command line non-JVM Apache Kafka producer and consumer* which can be easily installed.

If you don't have your own httpd logs available, you can use some freely available log files from NASA-HTTP web site access:

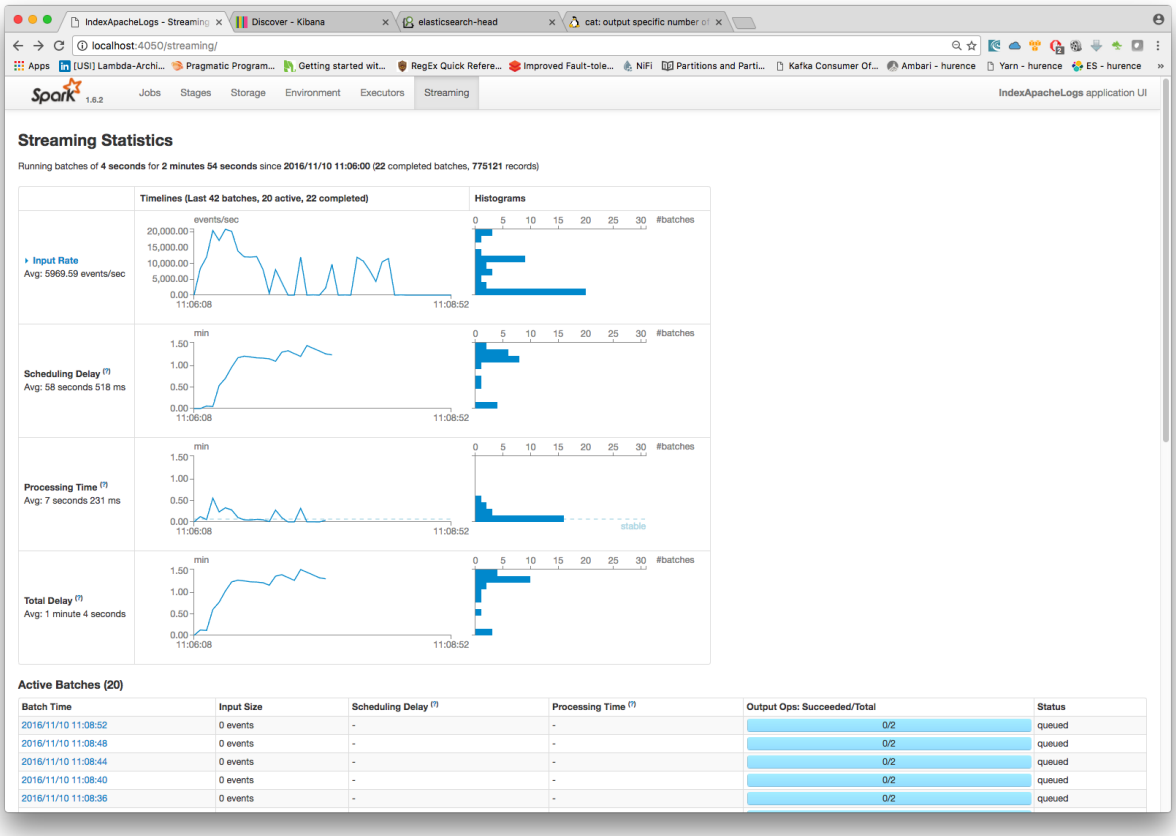
- Jul 01 to Jul 31, ASCII format, 20.7 MB gzip compressed
- Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed

Let's send the first 500000 lines of NASA http access over July 1995 to LogIsland with kafkacat to `logisland_raw` Kafka topic

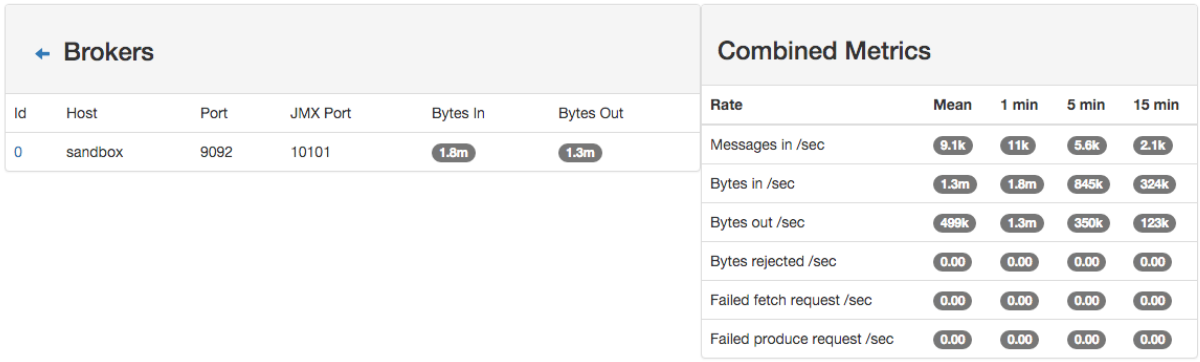
```
cd /tmp
wget ftp://ita.ee.lbl.gov/traces/NASA_access_log_Jul95.gz
gunzip NASA_access_log_Jul95.gz
head -500000 NASA_access_log_Jul95 | kafkacat -b sandbox:9092 -t logisland_raw
```

4. Monitor your spark jobs and Kafka topics

Now go to http://sandbox:4050/streaming/ to see how fast Spark can process your data



Another tool can help you to tweak and monitor your processing http://sandbox:9000/



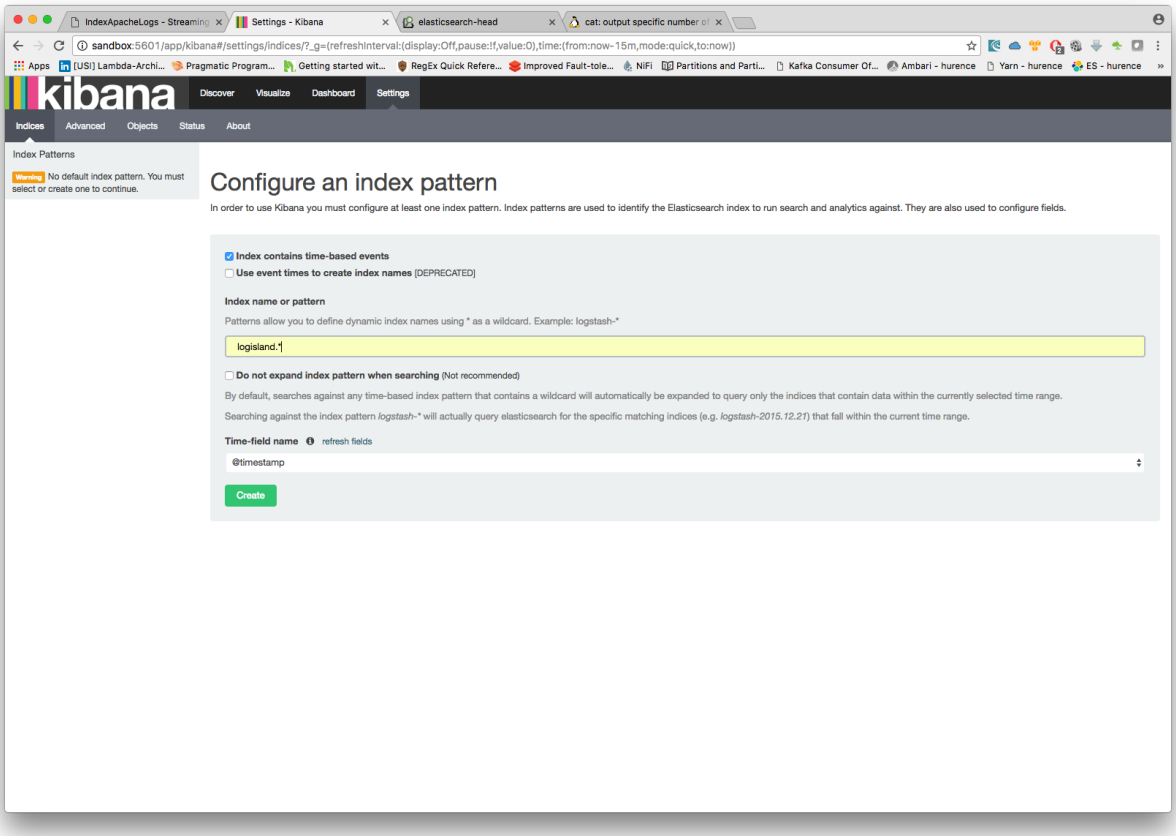
5. Inspect the logs

Kibana

With ElasticSearch, you can use Kibana.

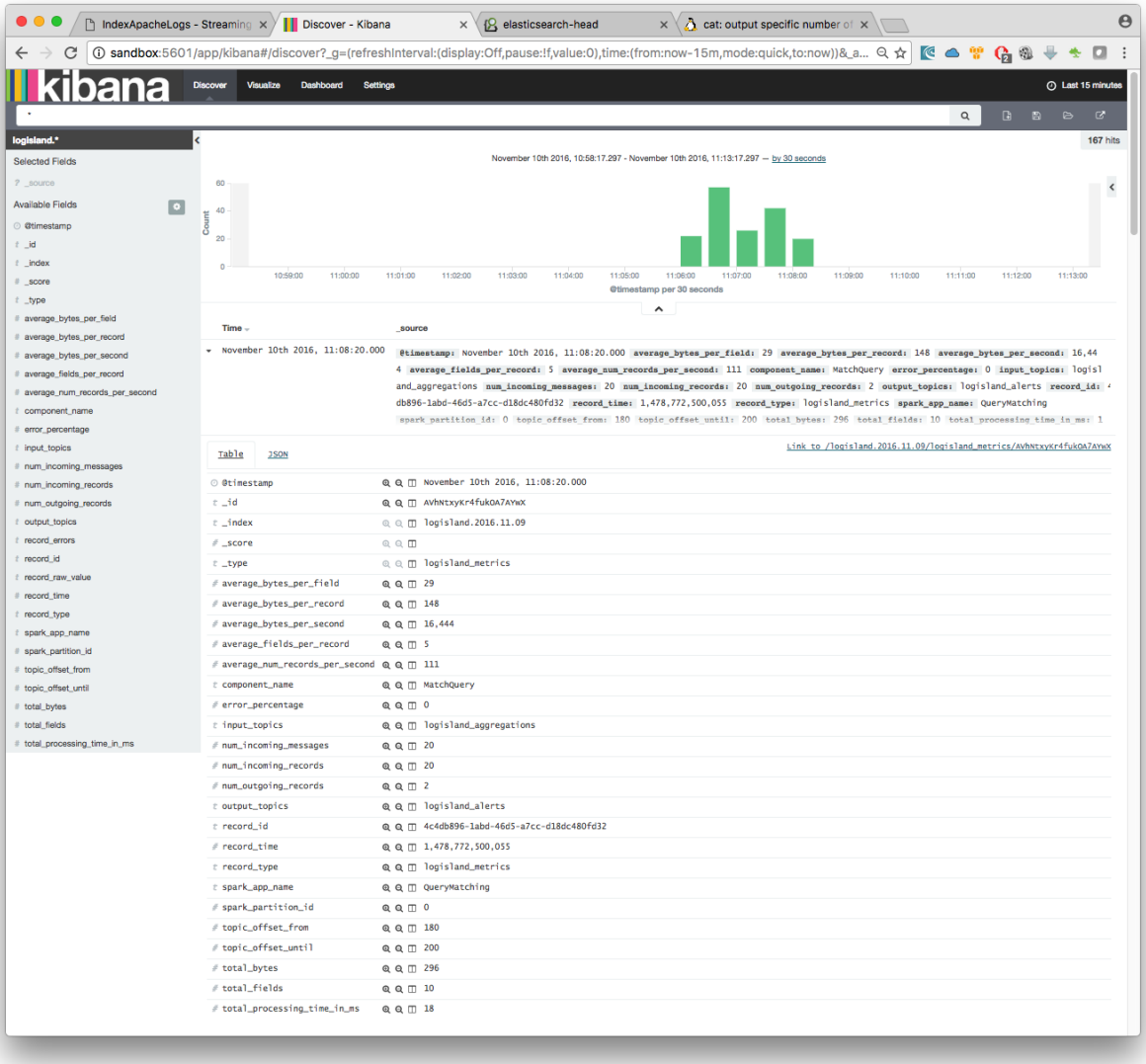
Open up your browser and go to http://sandbox:5601/ and you should be able to explore your apache logs.

Configure a new index pattern with `logisland.*` as the pattern name and `@timestamp` as the time value field.

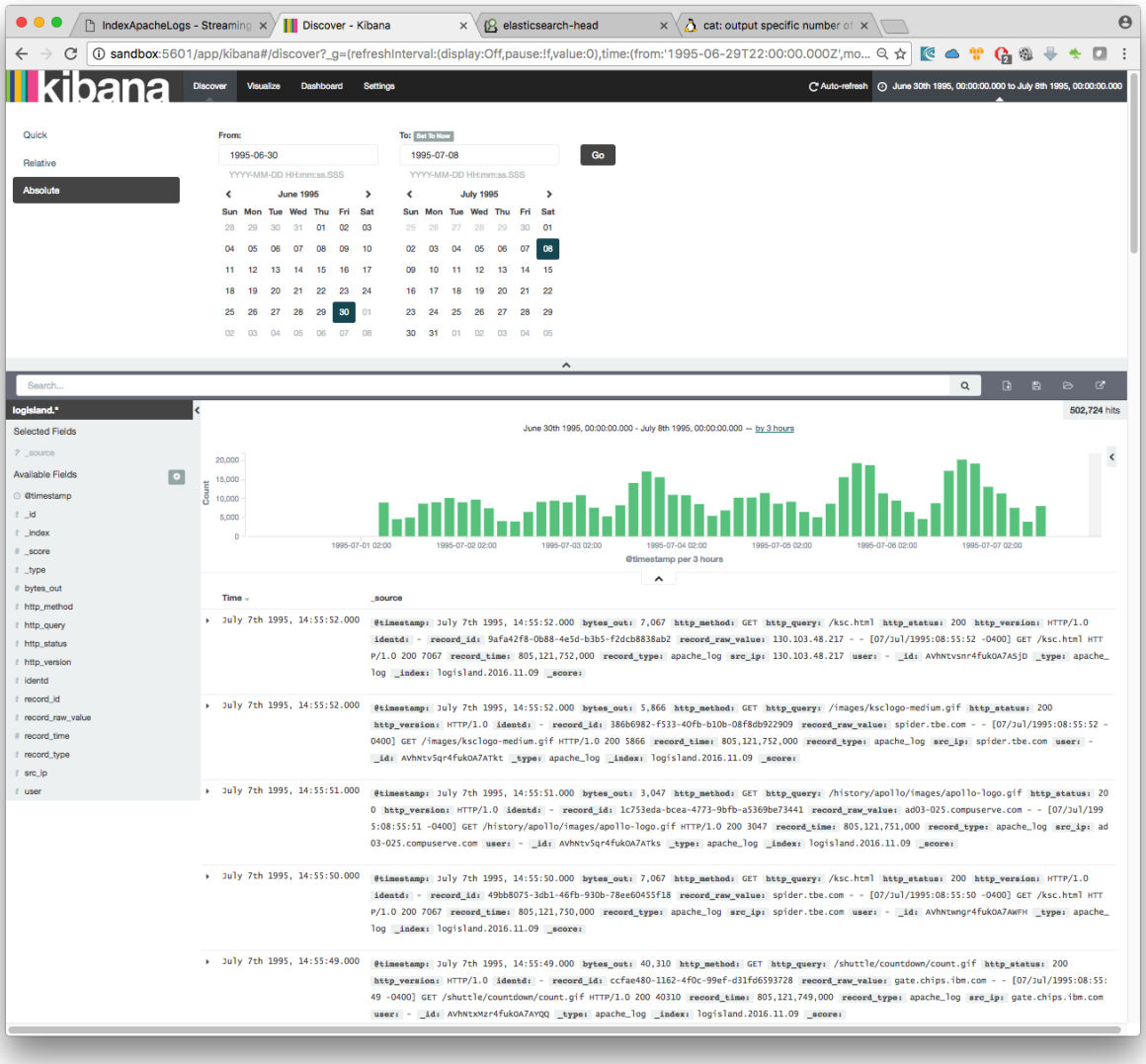


Then if you go to Explore panel for the latest 15' time window you'll only see logisland process_metrics events which

give you insights about the processing bandwidth of your streams.



As we explore data logs from july 1995 we'll have to select an absolute time filter from 1995-06-30 to 1995-07-08 to see the events.

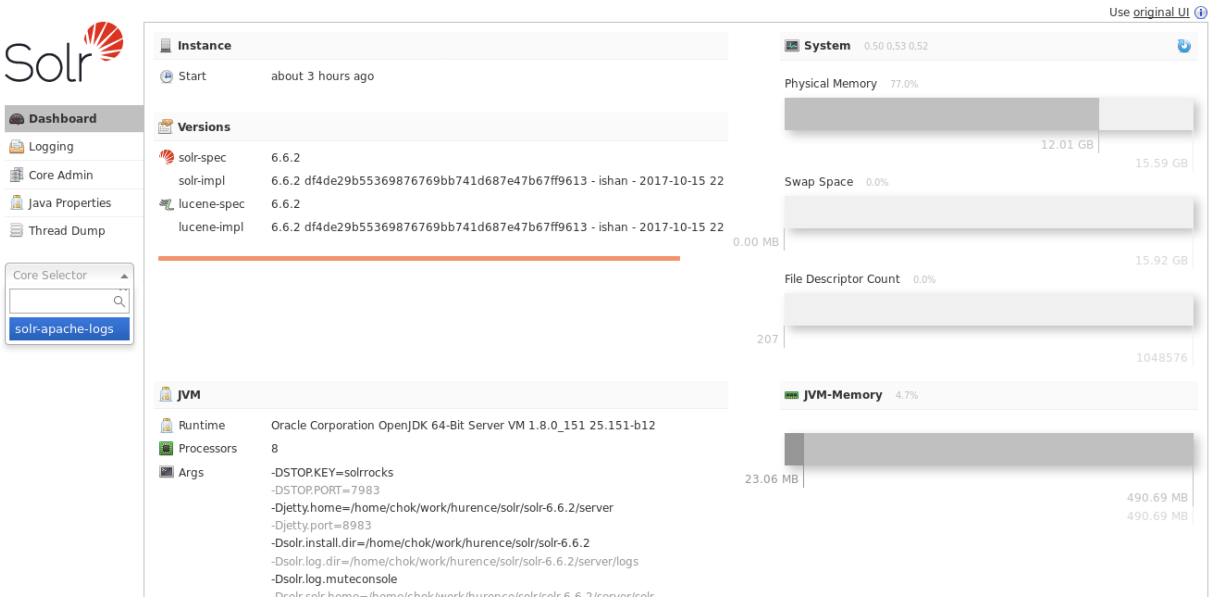


Solr

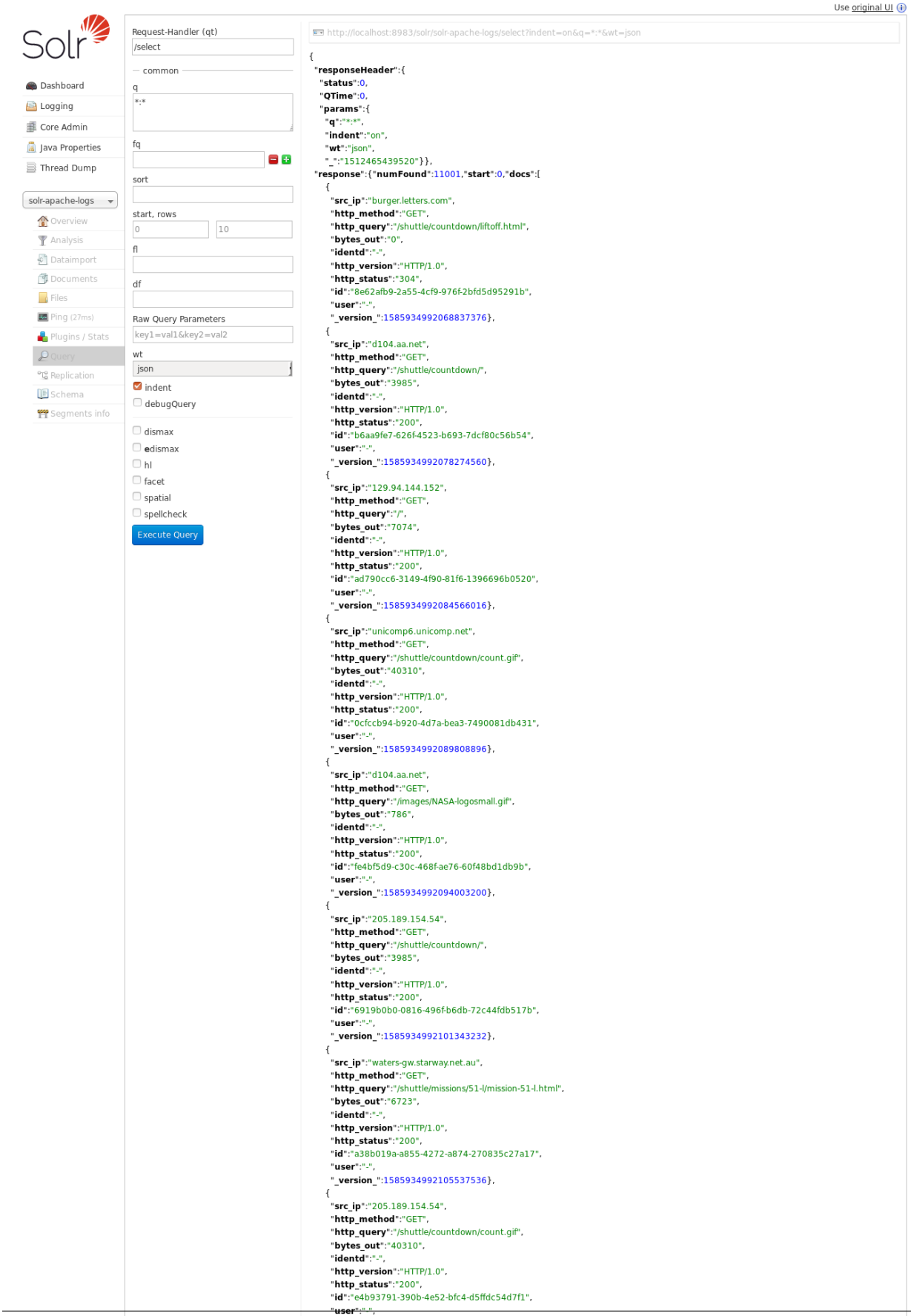
With Solr, you can directly use the solr web ui.

Open up your browser and go to http://sandbox:8983/solr and you should be able to view your apache logs.

In non cloud mode, use the core selector, to select the core `solr-apache-logs` :



Then, go to query and by clicking to Execute Query, you will see some data from your Apache logs :




```
{
 "responseHeader":{
  "status":0,
  "QTime":0,
  "params":{
   "q":"*:*",
   "indent":"on",
   "wt":"json",
   "_":"1512465439520"}},
 "response":{"numFound":11001,"start":0,"docs":[
   {
    "src_ip":"burger.letters.com",
    "http_method":"GET",
    "http_query":"/shuttle/countdown/liftoff.html",
    "bytes_out":"0",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"304",
    "id":"8e62afb9-2a55-4cf9-976f-2bfd5d95291b",
    "user":"-",
    "_version_":1585934992068837376},
   {
    "src_ip":"d104.aa.net",
    "http_method":"GET",
    "http_query":"/shuttle/countdown/",
    "bytes_out":"3985",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"200",
    "id":"b6aa9fe7-626f-4523-b693-7dcf80c56b54",
    "user":"-",
    "_version_":1585934992078274560},
   {
    "src_ip":"129.94.144.152",
    "http_method":"GET",
    "http_query":"/",
    "bytes_out":"7074",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"200",
    "id":"ad790cc6-3149-4f90-81f6-1396696b0520",
    "user":"-",
    "_version_":1585934992084566016},
   {
    "src_ip":"unicomp6.unicomp.net",
    "http_method":"GET",
    "http_query":"/shuttle/countdown/count.gif",
    "bytes_out":"40310",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"200",
    "id":"0cfccb94-b920-4d7a-bea3-7490081db431",
    "user":"-",
    "_version_":1585934992089808896},
   {
    "src_ip":"d104.aa.net",
    "http_method":"GET",
    "http_query":"/images/NASA-logosmall.gif",
    "bytes_out":"786",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"200",
    "id":"fe4bf5d9-c30c-468f-ae76-60f48bd1db9b",
    "user":"-",
    "_version_":1585934992094003200},
   {
    "src_ip":"205.189.154.54",
    "http_method":"GET",
    "http_query":"/shuttle/countdown/",
    "bytes_out":"3985",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"200",
    "id":"6919b0b0-0816-496f-b6db-72c44fdb517b",
    "user":"-",
    "_version_":1585934992101343232},
   {
    "src_ip":"waters-gw.starway.net.au",
    "http_method":"GET",
    "http_query":"/shuttle/missions/51-l/mission-51-l.html",
    "bytes_out":"6723",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"200",
    "id":"a38b019a-a855-4272-a874-270835c27a17",
    "user":"-",
    "_version_":1585934992105537536},
   {
    "src_ip":"205.189.154.54",
    "http_method":"GET",
    "http_query":"/shuttle/countdown/count.gif",
    "bytes_out":"40310",
    "identd":"-",
    "http_version":"HTTP/1.0",
    "http_status":"200",
    "id":"e4b93791-390b-4e52-bfc4-d5ffdc54d7f1",
    "user":"-",
    "_version_":1585934992110780416},
   {
    "src_ip":"unicomp6.unicomp.net",
    "http_method":"GET",
    "http_query":"/shuttle/countdown/",
    "bytes_out":"3985",
    "identd":"-",
    "http_version":"HTTP/1.0",
```

1.2.21 IIoT with OPC and Logisland

In this tutorial we'll show you how to ingest IIoT data from an OPC-UA server and process it with Logisland, storing everything into an elasticsearch database.

In particular, we'll use the Prosys OPC-UA simulation server you can download for free here

---

Note: You will need to have a logisland Docker environment. Please follow the prerequisites section for more information.

Please also remember to always turn on the simulation server before running the logisland job.

1.Install required components

For this tutorial please make sure to already have installed elasticsearch and OPC modules. If not you can just do it through the componentes.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_5_4_0-
↪client:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-connector-opc:1.1.2
```

2. Logisland job setup

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here for ElasticSearch :

```
docker exec -i -t logisland vim conf/opc-iiot.yml
```

We will start by explaining each part of the config file.

The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode with 1 cpu cores and 512M of RAM.

```
engine:
  component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
  type: engine
  documentation: Index some OPC-UA tagw with Logisland
  configuration:
  spark.app.name: OpcUaLogisland
  spark.master: local[2]
  spark.driver.memory: 512M
  spark.driver.cores: 1
  spark.executor.memory: 512M
  spark.executor.instances: 4
  spark.executor.cores: 1
  spark.yarn.queue: default
  spark.yarn.maxAppAttempts: 4
  spark.yarn.am.attemptFailuresValidityInterval: 1h
  spark.yarn.max.executor.failures: 20
  spark.yarn.executor.failuresValidityInterval: 1h
  spark.task.maxFailures: 8
```

(continues on next page)

(continued from previous page)

```
  spark.serializer: org.apache.spark.serializer.KryoSerializer
  spark.streaming.batchDuration: 3000
  spark.streaming.backpressure.enabled: false
  spark.streaming.blockInterval: 500
  spark.streaming.kafka.maxRatePerPartition: 10000
  spark.streaming.timeout: -1
  spark.streaming.unpersist: false
  spark.streaming.kafka.maxRetries: 3
  spark.streaming.ui.retainedBatches: 200
  spark.streaming.receiver.writeAheadLog.enable: false
  spark.ui.port: 4040
```

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job.

Here we have the OPC-UA source with all the connection parameters.

```
- controllerService: kc_source_service
  component: com.hurence.logisland.stream.spark.provider.
↪KafkaConnectStructuredSourceProviderService
    documentation: Kafka connect OPC-UA source service
    type: service
    configuration:
      kc.connector.class: com.hurence.logisland.connect.opc.ua.OpcUaSourceConnector
      kc.data.value.converter: com.hurence.logisland.connect.converter.
↪LogIslandRecordConverter
      kc.data.value.converter.properties: |
        record.serializer=com.hurence.logisland.serializer.KryoSerializer
      kc.data.key.converter.properties: |
        schemas.enable=false
      kc.data.key.converter: org.apache.kafka.connect.storage.StringConverter
      kc.worker.tasks.max: 1
      kc.connector.offset.backing.store: memory
      kc.connector.properties: |
        session.publicationRate=PT1S
        connection.socketTimeoutMillis=10000
        server.uri=opc.tcp://localhost:53530/OPCUA/SimulationServer
        tags.id=ns=5;s=Sawtooth1
        tags.sampling.rate=PT0.5S
        tags.stream.mode=SUBSCRIBE
```

In particular, we have

- A tag to be read: *"ns=5;s=Sawtooth1"*
- The tag will be subscribed and sampled each 0.5s
- The data will be published by the opc server each second (*session.publicationRate*)
- Please use your own opc server uri, in our case opc.tcp://localhost:53530/OPCUA/SimulationServer

Full connector documentation is on javadoc of class `com.hurence.logisland.connect.opc.ua.OpcUaSourceConnector`

Then we also define her Elasticsearch service that will be used later in the `BulkAddElasticsearch` processor.

```
- controllerService: elasticsearch_service
  component: com.hurence.logisland.service.elasticsearch.Elasticsearch_5_4_0_
↪ClientService
```

(continues on next page)

(continued from previous page)

```
  type: service
  documentation: elasticsearch service
  configuration:
    hosts: ${ES_HOSTS}
    cluster.name: ${ES_CLUSTER_NAME}
    batch.size: 5000
```

Inside this engine you will run a spark structured stream, taking records from the previously defined source and letting data flow through the processing pipeline till the console output.

```
- stream: ingest_stream
    component: com.hurence.logisland.stream.spark.structured.StructuredStream
    configuration:
      read.topics: /a/in
      read.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
      read.topics.key.serializer: com.hurence.logisland.serializer.StringSerializer
      read.stream.service.provider: kc_source_service
      write.topics: /a/out
      write.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
      write.topics.key.serializer: com.hurence.logisland.serializer.StringSerializer
      write.stream.service.provider: console_service
```

And now it's time to describe the parsing pipeline.

First, we need to extract the record thanks to a `FlatMap` processor

```
- processor: flatten
  component: com.hurence.logisland.processor.FlatMap
  type: processor
  documentation: "extract from root record"
  configuration:
    keep.root.record: false
    copy.root.record.fields: true
```

Now that the record is well-formed, we want to set the record time to be the same of the one given by the source (and stored on the field *tag_sampled_timestamp*).

For this, we use a `NormalizeFields` processor.

```
- processor: rename_fields
  component: com.hurence.logisland.processor.NormalizeFields
  type: processor
  documentation: "set record time to tag server generation time"
  configuration:
    conflict.resolution.policy: overwrite_existing
    record_time: tag_sampled_timestamp
```

Then, the last processor will index our records into elasticsearch

```
# add to elasticsearch
- processor: es_publisher
  component: com.hurence.logisland.processor.elasticsearch.BulkAddElasticsearch
  type: processor
  documentation: a processor that trace the processed events
  configuration:
    elasticsearch.client.service: elasticsearch_service
    default.index: logisland
```

(continues on next page)

(continued from previous page)

```
    default.type: event
    timebased.index: yesterday
    es.index.field: search_index
    es.type.field: record_type
```

3. Launch the script

Just ensure the Prosys OPC-UA server is up and running and that on the *Simulation* tab the simulation is ticked.

Then you can use the docker-compose file **docker-compose-opc-iiot.yml** available in the tar gz assembly in conf directory.

Note: If your simulation server is hosted on local and the hostname is different from 'localhost'. For example if your server uri is 'opc.tcp://${hostname}:53530/OPCUA/SimulationServer'. You can add it to logisland container add a extra_hosts properties to logisland container in docker-compose file so that it is accessible from the container.

```
logisland:
  network_mode: host
  image: hurence/logisland:1.1.2
  command: tail -f bin/logisland.sh
  environment:
    ZK_QUORUM: localhost:2181
    ES_HOSTS: localhost:9300
    ES_CLUSTER_NAME: es-logisland
  extra_hosts:
    - "${hostname}:127.0.0.1"
```

Then you can execute:

```
docker exec -i -t logisland bin/logisland.sh --conf conf/opc-iiot.yml
```

Note: Be sure to have added your server uri in conf/opc-iiot.yml file.

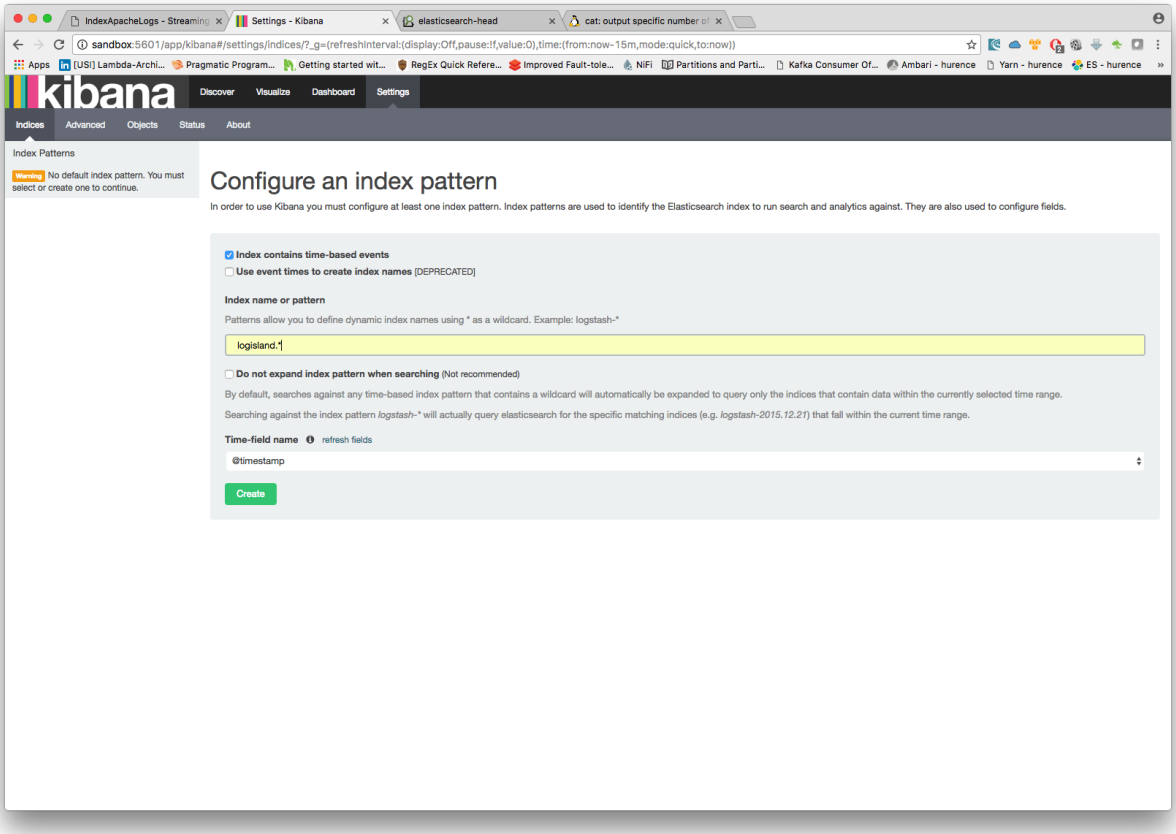
---

4. Inspect the records

With ElasticSearch, you can use Kibana.

Open up your browser and go to http://localhost:5601/ and you should be able to explore your apache logs.

Configure a new index pattern with `logisland.*` as the pattern name and `@timestamp` as the time value field.



Then if you go to Explore panel for the latest 15' time window you'll only see logisland process_metrics events which give you insights about the processing bandwidth of your streams.

1.2.22 Integrate Kafka Connect Sources & Sinks

In the following getting started tutorial, we'll focus on how to seamlessly integrate Kafka connect sources and sinks in logisland.

We can call this functionality *Logisland connect*.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

---

1. Logisland job setup

For this tutorial please make sure to already have installed elasticsearch and excel modules.

If not you can just do it through the components.sh command line:

```
bin/components.sh -i com.hurence.logisland:logisland-processor-elasticsearch:1.1.2

bin/components.sh -i com.hurence.logisland:logisland-service-elasticsearch_5_4_0-
↪client:1.1.2
```

(continues on next page)

(continued from previous page)

```
bin/components.sh -i com.github.jcustenborder.kafka.connect:kafka-connect-simulator:0.
↪1.118
```

The logisland job for this tutorial is already packaged in the tar.gz assembly and you can find it here for ElasticSearch :

```
docker exec -i -t logisland vim conf/logisland-kafka-connect.yml
```

We will start by explaining each part of the config file.

The engine

The first section configures the Spark engine (we will use a KafkaStreamProcessingEngine) to run in local mode.

```
engine:
  component: com.hurence.logisland.engine.spark.KafkaStreamProcessingEngine
  type: engine
  documentation: Use Kafka connectors with logisland
  configuration:
    spark.app.name: LogislandConnect
    spark.master: local[2]
    spark.driver.memory: 1G
    spark.driver.cores: 1
    spark.executor.memory: 2G
    spark.executor.instances: 4
    spark.executor.cores: 2
    spark.yarn.queue: default
    spark.yarn.maxAppAttempts: 4
    spark.yarn.am.attemptFailuresValidityInterval: 1h
    spark.yarn.max.executor.failures: 20
    spark.yarn.executor.failuresValidityInterval: 1h
    spark.task.maxFailures: 8
    spark.serializer: org.apache.spark.serializer.KryoSerializer
    spark.streaming.batchDuration: 1000
    spark.streaming.backpressure.enabled: false
    spark.streaming.unpersist: false
    spark.streaming.blockInterval: 500
    spark.streaming.kafka.maxRatePerPartition: 3000
    spark.streaming.timeout: -1
    spark.streaming.unpersist: false
    spark.streaming.kafka.maxRetries: 3
    spark.streaming.ui.retainedBatches: 200
    spark.streaming.receiver.writeAheadLog.enable: false
    spark.ui.port: 4050
```

The *controllerServiceConfigurations* part is here to define all services that be shared by processors within the whole job.

The parsing stream

Here we are going to use a special processor (`KafkaConnectStructuredSourceProviderService`) to use the kafka connect source as input for the structured stream defined below.

For this example, we are going to use the source *com.github.jcustenborder.kafka.connect.simulator.SimulatorSourceConnector* that generates records containing fake personal data at rate of 100 messages/s.

```
# Our source service
- controllerService: kc_source_service
  component: com.hurence.logisland.stream.spark.provider.
↪KafkaConnectStructuredSourceProviderService
  documentation: A kafka source connector provider reading from its own source and␣
↪providing structured streaming to the underlying layer
  configuration:
    # We will use the logisland record converter for both key and value
    kc.data.value.converter: com.hurence.logisland.connect.converter.
↪LogIslandRecordConverter
    # Use kryo to serialize the inner data
    kc.data.value.converter.properties: |
      record.serializer=com.hurence.logisland.serializer.KryoSerializer

    kc.data.key.converter: com.hurence.logisland.connect.converter.
↪LogIslandRecordConverter
    # Use kryo to serialize the inner data
    kc.data.key.converter.properties: |
      record.serializer=com.hurence.logisland.serializer.KryoSerializer
    # Only one task to handle source input (unique)
    kc.worker.tasks.max: 1
    # The kafka source connector to wrap (here we're using a simulator source)
    kc.connector.class: com.github.jcustenborder.kafka.connect.simulator.
↪SimulatorSourceConnector
    # The properties for the connector (as per connector documentation)
    kc.connector.properties: |
      key.schema.fields=email
      topic=simulator
      value.schema.fields=email,firstName,middleName,lastName,telephoneNumber,
↪dateOfBirth
    # We are using a standalone source for testing. We can store processed offsets in␣
↪memory
    kc.connector.offset.backing.store: memory
```

Note: The parameter **kc.connector.properties** contains the connector properties as you would have defined if you were using vanilla kafka connect.

As well, we are using a *memory* offset backing store. In a distributed scenario, you may have chosen a *kafka* topic based one.

Since each stream can be read and written, we are going to define as well a Kafka topic sink (`KafkaStructuredStreamProviderService`) that will be used as output for the structured stream defined below.

```
# Kafka sink configuration
- controllerService: kafka_out_service
  component: com.hurence.logisland.stream.spark.structured.provider.
↪KafkaStructuredStreamProviderService
  configuration:
    kafka.output.topics: logisland_raw
    kafka.error.topics: logisland_errors
    kafka.input.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    kafka.output.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
```

(continues on next page)

(continued from previous page)

```
    kafka.error.topics.serializer: com.hurence.logisland.serializer.JsonSerializer
    kafka.metadata.broker.list: sandbox:9092
    kafka.zookeeper.quorum: sandbox:2181
    kafka.topic.autoCreate: true
    kafka.topic.default.partitions: 4
    kafka.topic.default.replicationFactor: 1
```

So that, we can now define the *parsing stream* using those source and sink

```
######### parsing stream #############
- stream: parsing_stream_source
  component: com.hurence.logisland.stream.spark.structured.StructuredStream
  documentation: "Takes records from the kafka source and distributes related␣
↪partitions over a kafka topic. Records are then handed off to the indexing stream"
  configuration:
    read.topics: /a/in
    read.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.stream.service.provider: kc_source_service
    write.topics: logisland_raw
    write.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.stream.service.provider: kafka_out_service
```

Within this stream, a `FlatMap` processor takes out the value and key (required when using *StructuredStream* as source of records)

```
processorConfigurations:
 - processor: flatten
   component: com.hurence.logisland.processor.FlatMap
   type: processor
   documentation: "Takes out data from record_value"
   configuration:
     keep.root.record: false
     copy.root.record.fields: true
```

The indexing stream

Inside this engine, you will run a Kafka stream of processing, so we set up input/output topics and Kafka/Zookeeper hosts. Here the stream will read all the logs sent in `logisland_raw` topic and push the processing output into `logisland_events` topic.

Note: We want to specify an Avro output schema to validate our output records (and force their types accordingly). It's really for other streams to rely on a schema when processing records from a topic.

We can define some serializers to marshall all records from and to a topic.

```
- stream: parsing_stream_source
  component: com.hurence.logisland.stream.spark.structured.StructuredStream
  documentation: "Takes records from the kafka source and distributes related␣
↪partitions over a kafka topic. Records are then handed off to the indexing stream"
  configuration:
    read.topics: /a/in
```

(continues on next page)

(continued from previous page)

```
    read.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    read.stream.service.provider: kc_source_service
    write.topics: logisland_raw
    write.topics.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.topics.key.serializer: com.hurence.logisland.serializer.KryoSerializer
    write.stream.service.provider: kafka_out_service
```

Within this stream, a `DebugStream` processor takes a log line as a String and computes a `Record` as a sequence of fields.

```
processorConfigurations:
 # We just print the received records (but you may do something more interesting!)
 - processor: stream_debugger
   component: com.hurence.logisland.processor.DebugStream
   type: processor
   documentation: debug records
   configuration:
     event.serializer: json
```

This stream will process log entries as soon as they will be queued into *logisland_raw* Kafka topics, each log will be printed in the console and pushed back to Kafka in the `logisland_events` topic.

2. Launch the script

Connect a shell to your logisland container to launch the following streaming jobs.

```
docker exec -i -t logisland bin/logisland.sh --conf conf/logisland-kafka-connect.yml
```

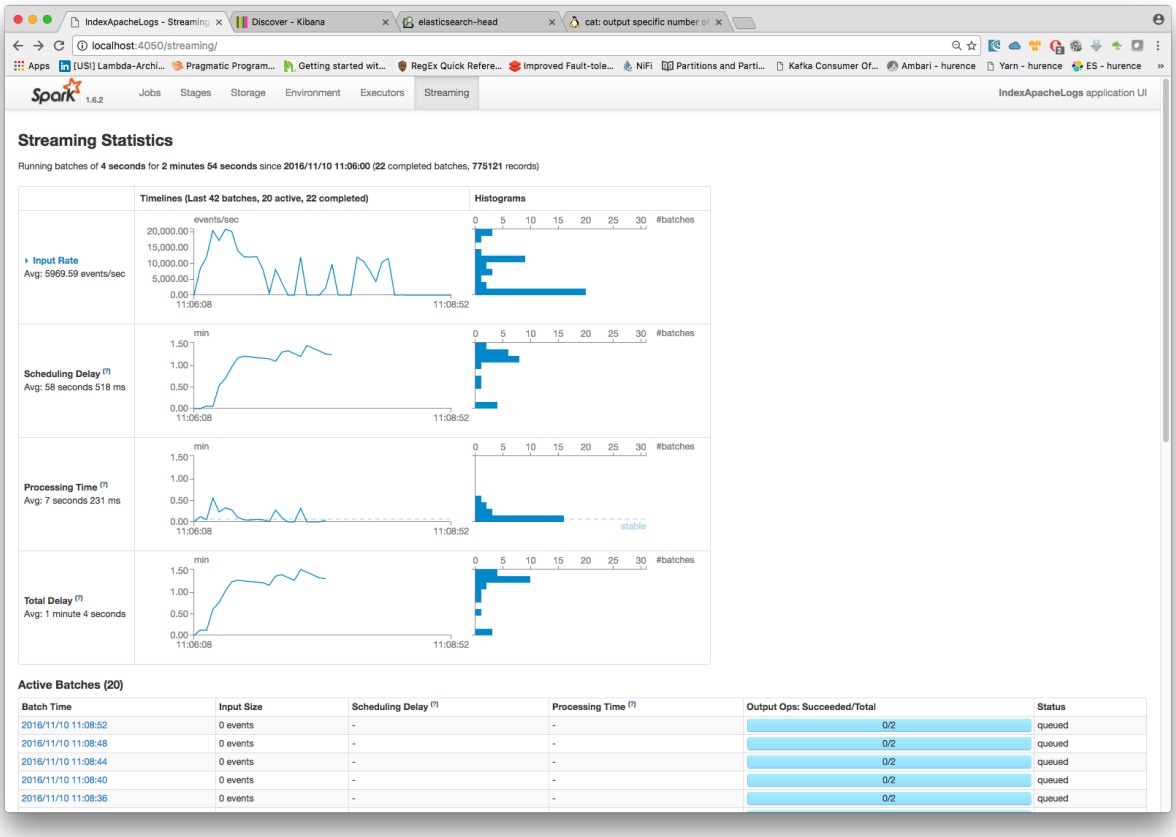
3. Examine your console output

Since we put a *DebugStream* processor, messages produced by our source connectors are then output to the console in json.

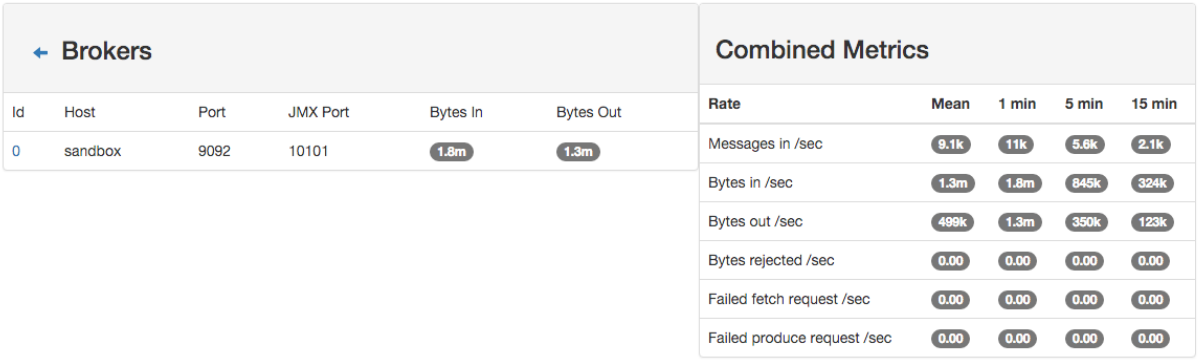
```
18/04/06 11:17:06 INFO DebugStream: {
 "id" : "9b17a9ac-97c4-44ef-9168-d298e8c53d42",
 "type" : "kafka_connect",
 "creationDate" : 1.4.106216376,
 "fields" : {
   "record_id" : "9b17a9ac-97c4-44ef-9168-d298e8c53d42",
   "firstName" : "London",
   "lastName" : "Marks",
   "telephoneNumber" : "005-694-4540",
   "record_key" : {
     "email" : "londonmarks@fake.com"
   },
   "middleName" : "Anna",
   "dateOfBirth" : 836179200000,
   "record_time" : 1.4.106216376,
   "record_type" : "kafka_connect",
   "email" : "londonmarks@fake.com"
 }
}
```

4. Monitor your spark jobs and Kafka topics

Now go to http://sandbox:4050/streaming/ to see how fast Spark can process your data



Another tool can help you to tweak and monitor your processing http://sandbox:9000/



1.2.23 Index JDBC messages

In the following getting started tutorial, we'll explain you how to read messages from a JDBC table.

The JDBC data will leverage the JDBC connector available as part of logisland connect.

Note: Be sure to know of to launch a logisland Docker environment by reading the prerequisites section

For kafka connect related information please follow as well the connectors section.

1.Install required components

For this tutorial please make sure to already have installed the kafka connect jdbc connector.

If not you can just do it through the componentes.sh command line:

```
bin/components.sh -r com.hurence.logisland.repackaged:kafka-connect-jdbc:5.0.0
```

2. Installing H2 datatabase

In this tutorial we'll use H2 Database.

H2 is a Java relational database

- Very fast database engine
- Open source
- Written in Java
- Supports standard SQL, JDBC API
- Embedded and Server mode, Clustering support
- Strong security features
- The PostgreSQL ODBC driver can be used
- Multi version concurrency

first wee need an sql engine. Let's use an **'H2 Java database<http://h2database.com/html/main.html>'_**. You can get the jar from their website and copy it to logisland lib folder inside Docker container. Then run the server on 9999 port

```
docker cp ./h2-1.4.197.jar logisland:/opt/logisland-1.1.2/lib
docker exec logisland java -jar lib/h2-1.4.197.jar   -webAllowOthers -tcpAllowOthers -
↪tcpPort 9999
```

You can manage your database through the web ui at http://sandbox:8082

With the URL JDBC parameter set to *jdbc:h2:tcp://sandbox:9999/~/test* you should be able to connect and create the following table

```
CREATE SCHEMA IF NOT EXISTS logisland;
USE logisland;

DROP TABLE IF EXISTS apache;

CREATE TABLE apache (record_id int auto_increment primary key, bytes_out integer,␣
↪http_method varchar(20), http_query varchar(200), http_status varchar(10), http_
↪version varchar(10), record_time timestamp, src_ip varchar(50), user varchar(20));
```

3. Logisland job setup

The interesting part in this tutorial is how to setup the JDBC stream.

Let's first focus on the stream configuration and then on its pipeline in order to extract the data in the right way.

Here we are going to use a special processor (`KafkaConnectStructuredSourceProviderService`) to use the kafka connect source as input for the structured stream defined below.

Logisland ships by default a kafka connect JDBC source implemented by the class *io.confluent.connect.jdbc.JdbcSourceConnector*.

You can find more information about how to configure a JDBC source in the official page of the JDBC Connector

Coming back to our example, we would like to read from a table called *logisland.apache* hosted in our local H2 database. The kafka connect controller service configuration will look like this:

```
- controllerService: kc_jdbc_source
  component: com.hurence.logisland.stream.spark.provider.
↪KafkaConnectStructuredSourceProviderService
  configuration:
    kc.data.value.converter: com.hurence.logisland.connect.converter.
↪LogIslandRecordConverter
    kc.data.value.converter.properties: |
      record.serializer=com.hurence.logisland.serializer.KryoSerializer
    kc.data.key.converter.properties:
    kc.data.key.converter: org.apache.kafka.connect.storage.StringConverter
    kc.worker.tasks.max: 1
    kc.partitions.max: 4
    kc.connector.class: io.confluent.connect.jdbc.JdbcSourceConnector
    kc.connector.offset.backing.store: memory
    kc.connector.properties: |
      connection.url=jdbc:h2:tcp://sandbox:9999/~/test
      connection.user=sa
      connection.password=
      mode=incrementing
      incrementing.column.name=RECORD_ID
      query=SELECT * FROM LOGISLAND.APACHE
      topic.prefix=test-jdbc-
```

Within this stream, a we need to extract the data coming from the JDBC.

First of all a `FlatMap` processor takes out the value and key (required when using *StructuredStream* as source of records)

```
processorConfigurations:
 - processor: flatten
   component: com.hurence.logisland.processor.FlatMap
   type: processor
   documentation: "Takes out data from record_value"
   configuration:
     keep.root.record: false
```

4. Launch the script

Now run the logisland job that will poll updates of new records inserted into *logisland.apache* table

```
docker exec logisland bin/logisland.sh --conf conf/index-jdbc-messages.yml
```

try to insert a few rows and have a look at the console output

```
INSERT into apache values (default, 46888, 'GET', '/shuttle/missions/sts-71/images/
↪KSC-95EC-0918.jpg', '200', 'HTTP/1.0', '2010-01-01 10:00:00' , 'net-1-141.eden.com',
↪ '-');
INSERT into apache values (default, 110,'GET','/cgi-bin/imagemap/countdown?99,176',
↪'302' ,'HTTP/1.0 ', '1995-07-01 04:01:06' ,'205.189.154.54', '-');
INSERT into apache values (default,12040,'GET','/shuttle/missions/sts-71/mission-sts-
↪71.html','200','HTTP/1.0', '1995-07-01 04:04:38','pme607.onramp.awinc.com', '-');
INSERT into apache values (default, 40310,'GET','/shuttle/countdown/count.gif','200' ,
↪'HTTP/1.0 ', '1995-07-01 04:05:18' ,'199.166.39.14', '-');
INSERT into apache values (default, 1.1.28,'GET','/images/dual-pad.gif','200' ,'HTTP/
↪1.0 ', '1995-07-01 04:04:10' ,'isdn6-34.dnai.com', '-');
INSERT into apache values (default,  9867,'GET','/software/winvn/winvn.html','200' ,
↪'HTTP/1.0 ', '1995-07-01 04:02:39' ,'dynip42.efn.org', '-');
INSERT into apache values (default, 1204,'GET','/images/KSC-logosmall.gif','200' ,
↪'HTTP/1.0 ', '1995-07-01 04:04:34' ,'netport-27.iu.net', '-');
```

it should be something like the following

```
...
18/09/04 12:47:33 INFO DebugStream: {
  "id" : "f7690b71-f339-4a84-8bd9-a0beb9ba5f92",
  "type" : "kafka_connect",
  "creationDate" : 1.4.165253831,
  "fields" : {
    "record_id" : "f7690b71-f339-4a84-8bd9-a0beb9ba5f92",
    "RECORD_TIME" : 0,
    "HTTP_STATUS" : "200",
    "SRC_IP" : "netport-27.iu.net",
    "RECORD_ID" : 7,
    "HTTP_QUERY" : "/images/KSC-logosmall.gif",
    "HTTP_VERSION" : "HTTP/1.0 ",
    "USER" : "-",
    "record_time" : 1.4.165253831,
    "record_type" : "kafka_connect",
    "HTTP_METHOD" : "GET",
    "BYTES_OUT" : 1204
  }
}
```

1.3 What's new in logisland ?

1.3.1 v1.4.0

- support for Azure databricks deployment (see -databricks and -chkloc run options)
- support for Azure Event Hubs through new structured stream source/sink service
- support for Avro serialization in structured streams
- support for opendistro (using elasticsearch 7.X service, validated against OD 1.4.0 with security [ssl - user/password])
- support for spark 2.4.0 through new logisland spark engine (warning: no support for 2.4.1 where scala needed version is 2.12, no more 2.11)

- added spark standalone experimental run mode (no documentation)

1.3.2 v1.3.0

- include Chronix timeseries api
- add EL support for FilterRecords
- add Solr8 support
- add Elasticsearch7 support

1.3.3 v1.1.2

- add a clock service
- improve monitoring
- improve Cassandra support

1.3.4 v1.0.0

- add support for JMS kafka connect source
- add support for JDBC kafka connect source
- add Cassandra datastore service
- support all Kafka connect sinks
- add KafkaStreams engine
- update documentation
- fix test framework (runner)
- added vanilla java engine

1.3.5 v0.14.0

- add support for SOLR
- add support for Chronix timeseries
- review Datastore API
- fix matchquery update field policy issue
- remove elasticsearch 2.3 support

1.3.6 v0.10.0

- add kibana pcap panel cyber-security feature gui #187
- add support for elasticsearch 2.4 feature processor
- add support for elasticsearch 5 feature processor #214
- fix pb in kafkaStreamProcessingEngine (2.1) #244

- allow to set a default profile during build #271
- add ElasticSearch Service feature framework #241
- add multiGet elastic search processor feature processor #255
- fix Pcap telemetry processor issue #180 #224
- Make build work if no profile specified (use the highest hdp one) build #210
- implement Logisland agent #201
- fix travis build randomly fails on travis CI (spark-engine module tests) bug framework #159
- support maven profiles to handle dépendencies (hdp 2.4 & hdp 2.5) #116
- add a RESTful API for components live update agent feature framework #42
- add a logisland agent agent enhancement feature framework #117
- add a Topic metadata view feature gui #101
- add scheduler view feature framework gui #103
- add job configuration view feature gui #94
- add a global logisland.properties agent feature #71
- add a Topic metadata registry feature framework
- integrate BRO files & notification through a BroProcessor feature processor security #93
- add Support for SMTP/Mailer Processor feature processor security #138
- add a Release/deployment documentation #108
- Ensure source files have a licence header
- add HBase service to get and scan records
- add Multiget elasticsearch enricher processor
- add sessionization processor
- improve topic management in web ui gui #222
- Docker images shall be builded automatically framework #200
- fix classpath issue bug framework #247
- add Netflow telemetry Processor cyber-security feature processor #181
- add an “How to contribute page” documentation #183
- fix PutElasticsearch throws UnsupportedOperationException when duplicate document is found bug processor #221
- Feature/maven docker#200 enhancement framework #242
- Feature/partitioner enhancement framework #238
- add PCAP telemetry Processor cyber-security feature processor #180
- Move Mailer Processor into commons plugins build #196
- Origin/webanalytics framework processor web-analytics #236
- rename Plugins to Processors in online documentation documentation #173

1.3.7 v0.9.8

- add a retry parameter to PutElasticsearch bug enhancement processor #124
- add Timezone managmt to SplitText enhancement processor #126
- add IdempotentId processor enhancement feature processor #127
- migrate to Kafka 0.9 enhancement

1.3.8 v0.9.7

- add HDFS burner feature processor #89
- add ExtractJsonPath processor #90
- check compatibility with HDP 2.5 #112
- sometimes the drivers fails with status SUCCEEDED which prevents YARN to resubmit the job automatically #105
- logisland crashes when starting with wrong offsets #111
- add type checking for SplitText component enhancement #46
- add optional regex to SplitText #106
- add record schema management with ConvertFieldsType processor #75
- add field auto extractor processor : SplitTextWithProperties #49
- add a new RemoveFields processor
- add a NormalizeFields processor #88
- Add notion of asserting the asserted fields in MockRecord

1.3.9 v0.9.6

- add a Documentation generator for plugins feature #69
- add SQL aggregator plugin feature #74
- #66 merge elasticsearch-shaded and elasticsearch-plugin enhancement
- #73 add metric aggregator processor feature
- #57 add sampling processor enhancement
- #72 integrate OutlierDetection plugin feature
- #34 integrate QueryMatcherProcessor bug

1.3.10 v0.9.5

- generify API from Event to Records
- add docker container for demo
- add topic auto-creation parameters
- add Record validators

- add processor chaining that works globally on an input/output topic and pipe in-memory contexts into sub-processors
- better error handling for SplitText
- testRunner API
- migrate LogParser to LogProcessor Interface
- reporting metrics to know where are exactly the processors on the topics
- add an HDFSBurner Engine
- yarn stability improvements
- more spark parameters handling
- driver failover through Zookeper offset checkpointing
- add raw_content to event if regex matching failed in SplitText
- integration testing with embedded Kafka/Spark
- processor chaining
-

CHAPTER 2

Indices and tables

- genindex
- modindex
- search